المملكـــة العربيــــة الســـــعودية

المؤسسة العامة للتعليم الفني والتدريب المهني

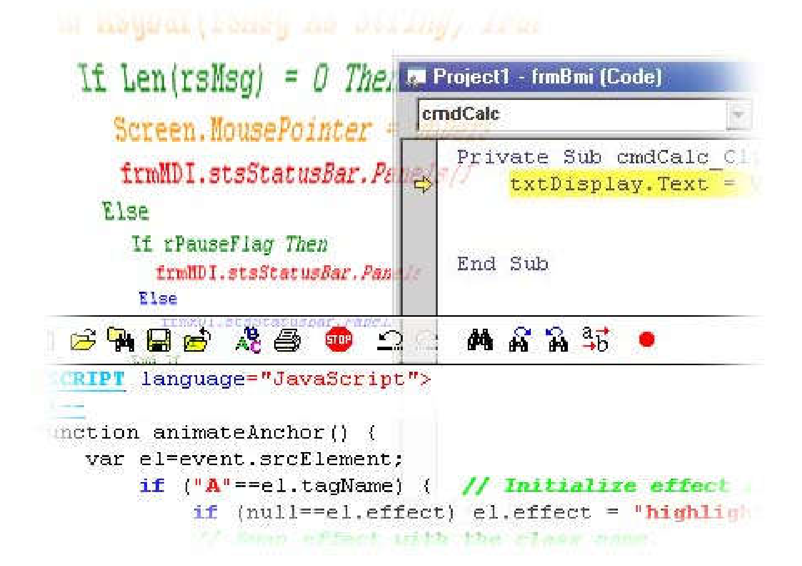
الإدارة العامة لتصميم وتطوير المناهج



برمجيات

تصميم قواعد البيانات

١٦٢ حاب



مقدمة

الحمد لله وحده، والصلاة والسلام على من لا نبي بعده، محمد وعلى آله وصحبه، وبعد:

تسعى المؤسسة العامة للتعليم الفني والتدريب المهني لتأهيل الكوادر الوطنية المدربة القادرة على شغل الوظائف التقنية والفنية والمهنية المتوفرة في سوق العمل، ويأتي هذا الاهتمام نتيجة للتوجهات السديدة من لدن قادة هذا الوطن التي تصب في مجملها نحو إيجاد وطن متكامل يعتمد ذاتياً على موارده وعلى قوة شبابه المسلح بالعلم والإيمان من أجل الاستمرار قدماً في دفع عجلة التقدم التنموي، لتصل بعون الله تعالى لمصاف الدول المتقدمة صناعياً.

وقد خطت الإدارة العامة لتصميم وتطوير المناهج خطوة إيجابية تتفق مع التجارب الدولية المتقدمة في بناء البرامج التدريبية، وفق أساليب علمية حديثة تحاكي متطلبات سوق العمل بكافة تخصصاته لتلبي متطلباته، وقد تمثلت هذه الخطوة في مشروع إعداد المعايير المهنية الوطنية الذي يمثل الركيزة الأساسية في بناء البرامج التدريبية، إذ تعتمد المعايير في بنائها على تشكيل لجان تخصصية تمثل سوق العمل والمؤسسة العامة للتعليم الفني والتدريب المهني بحيث تتوافق الرؤية العلمية مع الواقع العملي الذي تفرضه متطلبات سوق العمل، لتخرج هذه اللجان في النهاية بنظرة متكاملة لبرنامج تدريبي أكثر التصاقاً بسوق العمل، وأكثر واقعية في تحقيق متطلباته الأساسية.

وتتناول هذه الحقيبة التدريبية " تصميم قواعد البيانات" لمتدربي قسم" برمجيات " للكليات التقنية موضوعات حيوية تتناول كيفية اكتساب المهارات اللازمة لهذا التخصص.

والإدارة العامة لتصميم وتطوير المناهج وهي تضع بين يديك هذه الحقيبة التدريبية تأمل من الله عز وجل أن تسهم بشكل مباشر في تأصيل المهارات الضرورية اللازمة، بأسلوب مبسط يخلو من التعقيد، وبالاستعانة بالتطبيقات والأشكال التي تدعم عملية اكتساب هذه المهارات.

والله نسأل أن يوفق القائمين على إعدادها والمستفيدين منها لما يحبه ويرضاه، إنه سميع مجيب الدعاء.

الإدارة العامة لتصميم وتطوير المناهج

المملكة العربية السعودية
المؤسسة العامة للتعليم الفني والتدريب المهني
الإدارة العامة لتصميم وتطوير المناهج



تصميم قواعد البيانات

مقدمة لتصميم قواعد البيانات

مقدمة لتصميم قواعد البيانات

١

الجدارة:

القدرة على وصف مكونات نظام قاعدة البيانات ودورة حياة النظام

الأهداف:

١. أن يتعرف المتدرب مكونات نظام قاعدة البيانات

٢. أن يتعرف المتدرب دورة الحياة لنظام قاعدة البيانات

مستوى الأداء المطلوب:

أن يتقن المتدرب وصف مكونات النظام ومراحل تطويره بنسبة ١٠٠٪.

الوقت المتوقع للتدريب:

ساعتان

الوسائل المساعدة:

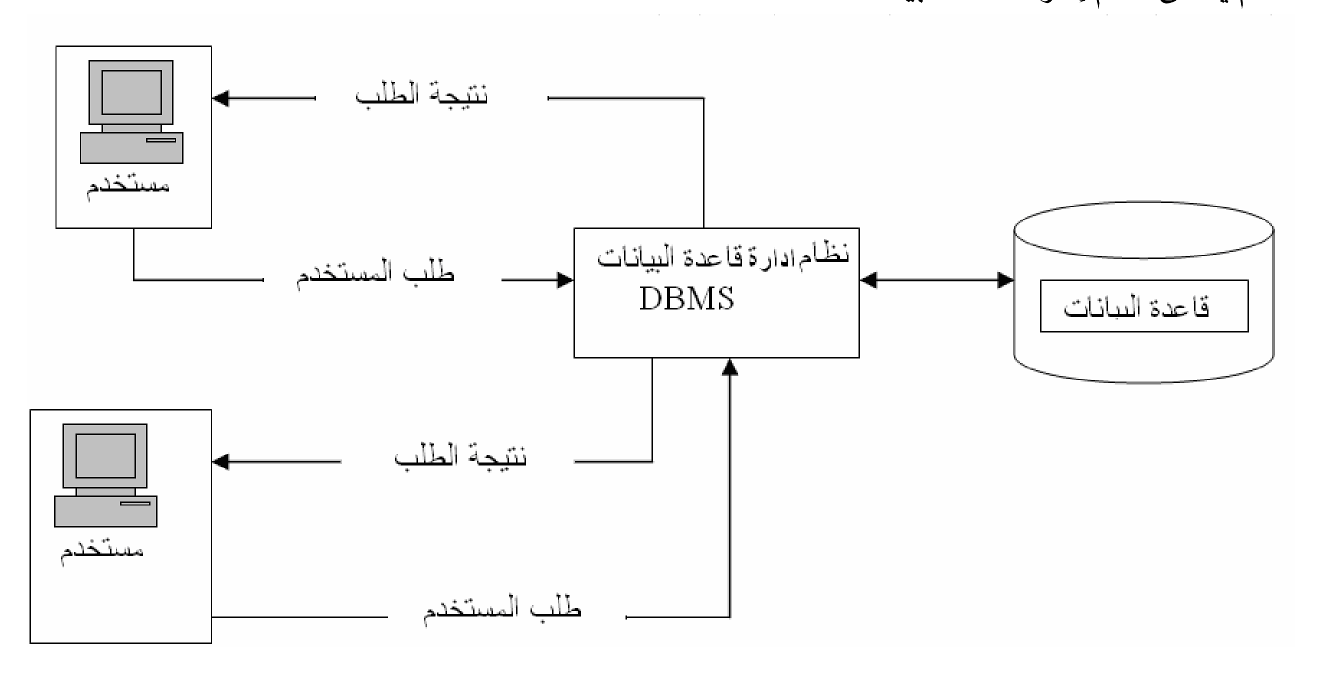
قلم + دفتر

متطلبات الجدارة:

أن يكون المتدرب قد أتقن جميع الجدارت في مقدمة قواعد البيانات .

تعريف قاعدة البيانات

قاعدة البيانات: هي عبارة عن مجموعة المعلومات والبيانات المخزنة بطريقة نموذجية ودون تكرار والمتصلة مع بعضها وفق علاقات متبادلة..ومن أمثلة قواعد البيانات نظام تسجيل المتدربين حيث يقوم على تخزين البيانات الخاصة بالمتدربين والمتدربين والمقررات والشعب... الخ في جداول. وكذلك تحديد العلاقات بين هذه الجداول وفق أسس محددة وثابتة تعتمد على قواعد العمل في هذا النظام وكذلك على استخدام الطرق الصحيحة في عملية تصميم قاعدة البيانات. وتكون قاعدة البيانات مفصولة عن البرامج والتطبيقات التي تقوم بمعالجة هذه البيانات مثل برامج الإدخال والتعديل والحذف ويدير قاعدة البيانات نظام يمسى نظام إدارة قاعدة البيانات.



تعريف نظام إدارة قاعدة البيانات

ما هي إدارة نظام قاعدة البيانات DBMS (Database Management Information System) ؟

هي عبارة عن مجموعة البرامج التي تدير وتتحكم بعملية تخزين واسترجاع البيانات، وتوفر كذلك إمكانية قيام عدد كبير من المستخدمين من الوصول والتعامل مع البيانات، وينظر إليها كذلك على أنها حلقة الوصل بين المستخدمين وقاعدة البيانات، بحيث تقوم باستقبال متطلبات المستخدمين ومن ثم نقلها إلى قاعدة البيانات وتنفيذ البرامج اللازمة لتنفيذ هذه المتطلبات ومن ثم تزويد المستخدم بالنتائج المطلوبة.

مكونات نظام قاعدة البيانات

يقسم نظام قاعدة البيانات إلى خمسة أقسام :

١ - **المكونات المادية** (**Hardware**) :

وتشمل جميع الأجهزة المادية في النظام مثل الحاسبات، الأجهزة الطرفية، الطابعات وكذلك أجهزة الاتصال في بيئة قاعد البيانات الموسعة... الخ.

٢ - **البرمجيات** (Software):

وهي مجموعة البرامج المستخدمة في قاعدة البيانات، وتقسم إلى ثلاثة أقسام:

أ - **أنظمة التشغيل**: وهي البرامج التي تقوم بإدارة الأجهزة وتهيئتها للعمل وتمكين بقية البرامج من العمل مثل Linux, Unix, Windows.....

ب - **برنامج قاعدة البيانات**: وهو البرنامج الذي يتولى إدارة قاعدة البيانات مثل Oracle, Sybase, DB2 ... .

ج - **البرامج التطبيقية والبرامج المساعدة**: وهي البرامج التي تقوم بعمليات الاسترجاع والتخزين وكذلك استخراج التقارير... .

٣ - **المستخدمون** : وهم عبارة عن الأشخاص الذين يقومون بالعمل في بيئة قاعدة البيانات وهم :

أ - مدير **النظام**: وهو الشخص المسؤول عن إدارة عمل البيئة العامة التي يعمل بها نظام قاعدة البيانات ويقوم بما يلي:

١. بإدارة المستخدمين ومنح الصلاحيات لاستخدام النظام.
٢. إدارة أجهزة التخزين والأجهزة الأخرى.
٣. متابعة عمل النظام.

ب - **مدير قاعدة البيانات**: وهو المسؤول عن إدارة قاعدة البيانات وتشمل واجباته:

١. تحديد متطلبات قاعدة البيانات من برامج وتجهيزات.
٢. متابعة نظام قاعدة البيانات وتنسيق عملية استخدامه.
٣. توفير الأمن والحماية للنظام.
٤. تصميم آليات المحافظة على قاعدة البيانات وتحديد الإجراءات اللازمة لتوفير الخدمات للمستخدمين الآخرين.

ج - **مصمم قاعدة البيانات** وهو الشخص(الأشخاص) الذي يقوم بعملية تصميم قاعدة البيانات وتشمل واجباته:

١. تحديد البيانات الواجب تخزينها في قاعدة البيانات
٢. تصميم أفضل التراكيب لحفظ البيانات .
٣. تصميم قاعدة بيانات خالية من التكرار .
٤. تحديد طرق الوصول والمعالجة والاسترجاع للبيانات من خلال تصميم الشاشات والتقارير الواجب استخدامها .
٥. توثيق عملية التصميم وطرق الوصول للبيانات .

د - **المبرمجون ومحللو النظم** : وهم الأشخاص الذين يقومون بعملية تصميم البرامج وتنفيذها وتشمل واجباتهم :

١. تصميم التطبيقات وتحويلها إلى برامج بلغة(لغات) برمجة حسب السياسات المقررة في عملية التصميم .
٢. تنفيذ وتطبيق تلك البرامج والتأكد من سلامتها .
٣. عمل الصيانة اللازمة لتلك البرامج .

هـ - **المستخدم النهائي**: وهو الشخص أو مجموعة الأشخاص الذين يقومون بالعمل اليومي على النظام وتطبيق البرامج في مجال محدد مثل الاسترجاع، التعديل، الحذف، تنفيذ التقارير... الخ .

٤ - **الإجراءات والعمليات**: وهي عبارة عن القوانين والتعليمات التي تحكم عمل قاعدة البيانات بشكل صحيح وتكون على شكل تعليمات موثقة بشكل واضح ومحدد.

٥ - **البيانات**: وهي أهم مكونات النظام حيث تشمل مجموعة الحقائق المخزنة في قاعدة البيانات. وكون البيانات تكون على شكل بدائي إذ لابد من تحديد مكان وكيفية التخزين لهذه البيانات حتى تسهل عملية معالجتها والاستفادة منها وهذا عمل المصمم .

أهمية تصميم قواعد البيانات :

إن عملية بناء قاعدة بيانات جيدة لا يأتي بتلك السهولة ، إذ لابد من بذل جهد كبير للحصول على قاعدة بيانات جيدة. والتصميم الجيد لقاعدة البيانات يسهل عملية استخدام وإدارة هذه القاعدة أما التصميم السيئ فسيؤدي إلى تكرار البيانات(ويعني وجود نفس البيانات في أكثر من مكان) وبالتالي تصعب عملية الحفاظ على توافقية البيانات وعادة ما يؤدي تكرار البيانات إلى نتائج غير صحيحة عند طلب تلك البيانات من تلك القاعدة وهذا بدوره يؤدي إلى أن أي قرارات إدارية وكذلك أي تخطيط مستقبلي سيكون خاطئا لاعتماده على معلومات غير صحيحة.

دورة الحياة لنظام قاعدة البيانات :

١ - **الدراسة المبدئية للنظام القائم وتشمل ما يلي:**

أ - تحليل الوضع الحالي للمؤسسة ومعرفة طبيعة الإجراءات المستخدمة والتعليمات وقواعد العمل .

ب -تحديد المشاكل التي تواجه النظام المستخدم وكذلك القيود المادية مثل الطاقة البشرية والتمويل المتوفر لتطوير أو استبدال النظام الحالي .

ج - تحديد الأهداف الواجب تحقيقها والمزايا المطلوبة في النظام الجديد.

٢ - **تصميم قاعدة البيانات**: وتعتبر هذه المرحلة من أهم المراحل في دورة حياة النظام إذ لابد من بذل جهد كبير لتصميم النظام للوصول إلى نظام جيد وتؤدي الأهداف المرجوة من عمل النظام وتشمل عملية التصميم ما يلي:

أ - بناء نموذج المفاهيم وتشمل هذه العملية عدة خطوات(سنتطرق إلى هذه العملية بالتفصيل في الفصول اللاحقة):

١. تحليل البيانات ومتطلبات المستخدمين والإجراءات المطلوبة
٢. تعريف وتحديد الكيانات وخصائصها وعلاقتها مع بعضها وكذلك وضعها في الصيغة المعيارية.
٣. رسم مخطط المفاهيم وهو عبارة عن نموذج رسومي يوصف كيانات النظام وعلاقتها مع بعضها.
٤. تعديل النموذج بحيث يشمل الإجراءات الرئيسة ، وقواعد عمليات الإضافة والتعديل والحذف على البيانات والتقارير ، والشاشات ، ومقدار التشاركية و توافقية البيانات.... .

ب - اختيار نظام إدارة قاعدة البيانات (DBMS).

ج - تحويل نموذج المفاهيم إلى نموذج داخلي بالاعتماد على نظام إدارة قاعدة البيانات(DBMS).

د - التصميم المادي وتتم خلاله عملية وضع مواصفات التخزين والوسائط المستخدمة في عملية التخزين وطرق الوصول للبيانات بالاعتماد على نظام إدارة قاعدة البيانات (DBMS).

٣ - **تنفيذ النظام:** وخلال هذه المرحلة تتم عملية إنشاء الجداول وكتابة جميع البرامج اللازمة لتنفيذ متطلبات النظام من الشاشات المختلفة و التقارير المطلوبة ... .

٤ - **عملية الفحص والتقييم للنظام وتشمل :**

أ - فحص قاعدة البيانات والتأكد من عملها بشكل صحيح.

ب -تقييم عمل البرامج والتطبيقات المستخدمة.

٥ - **تطبيق النظام في مكان العمل:** وتشمل هذه العملية عمليات إنشاء الجداول والمستخدمين والصلاحيات..، وتحميل جميع البرامج والتطبيقات وتنفيذها في البيئة الحقيقة التي يجب أن يعمل بها النظام.

٦ -**متابعة عمل النظام:** وهذه العملية تستمر طيلة فترة حياة النظام للتأكد من عمله بشكل صحيح وكذلك تعديل النظام ليتواءم مع المتطلبات الجديدة لبيئة العمل مثل تغير القوانين والأنظمة وقواعد العمل.

تمارين

١ - أي العبارات التالية صحيح وأيها خاطئة ؟

أ - من واجبات المبرمج توفير الأمن والحماية للنظام

ب -المستخدم النهائي هو الشخص الذي يقوم بتوثيق عملية التصميم وطرق الوصول للبيانات.

ج - تعتبر البيانات من أهم مكونات نظام قاعدة البيانات.

٢ - عرف ما يلي:

- قاعدة البيانات
- نظام إدارة قاعدة البيانات

٣ - ما هي أهمية تصميم قاعدة البيانات ؟

٤ - اشرح مرحلة تصميم قاعدة البيانات شرحا مفصلا .

٥ - اذكر واجبات كل من مصمم قاعدة البيانات، المبرمجين ومحللي النظم.

المملكة العربية السعودية
المؤسسة العامة للتعليم الفني والتدريب المهني
الإدارة العامة لتصميم وتطوير المناهج



تصميم قواعد البيانات

قواعد البيانات العلائقية

قواعد البيانات العلائقية

٢

الجدارة:

القدرة على وصف قواعد البيانات العلائقية وصفا صحيحا .

الأهداف:

- أن يتعرف المتدرب على قاعدة البيانات العلائقية .
- أن يميز المتدرب بين مختلف أنواع المفاتيح للجداول (العلاقة).
- أن يتعرف المتدرب على مختلف أنواع التشاركية بين الجداول (العلاقات).

مستوى الأداء المطلوب:

أن يصف المتدرب قواعد البيانات العلائقية وصفا صحيحا وكاملا بنسبة ١٠٠ ٪.

الوقت المتوقع للتدريب:

ساعتان.

الوسائل المساعدة:

قلم + دفتر.

متطلبات الجدارة:

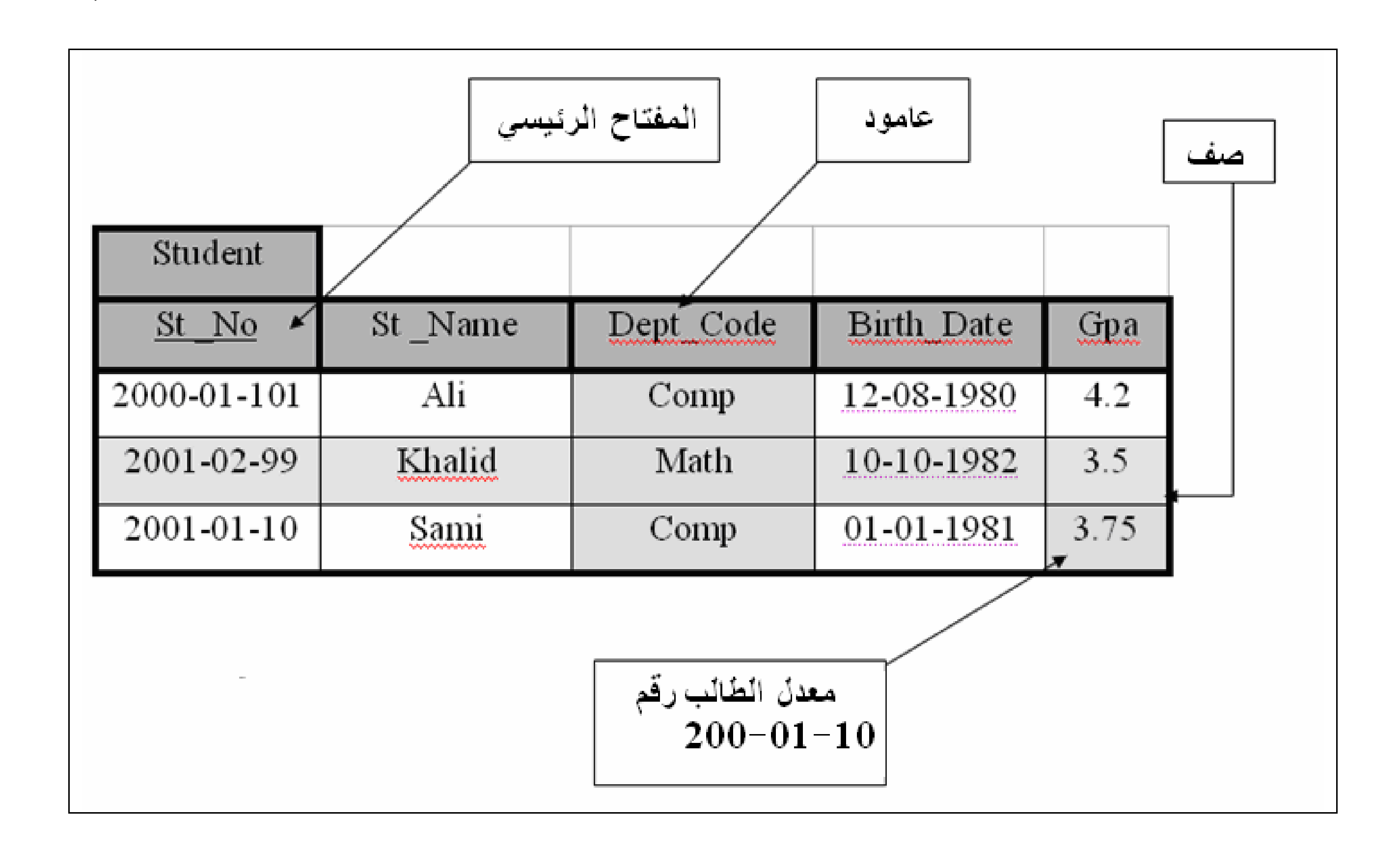
أن يكون المتدرب قد أتقن الجدارة في الوحدة السابقة.

قاعدة البيانات العلائقية :

بدأ نشوء مفهوم قواعد البيانات العلائقية عام ١٩٧٠ عندما قدم العالم Codd اقتراحا لهذا النموذج والذي تم بناؤه على **نظريات الجبر العلائقي** ومن هنا برزت قوة هذا النموذج وسرعة انتشاره فيما بعد. ففي مطلع الثمانينات بدأت الكثير من الشركات بتبني هذا النموذج وتطبيقه، فنلاحظ الآن أن معظم أنظمة قواعد البيانات الموجودة في الأسواق تتوافق مع هذا النموذج. وتتلخص فكرة النموذج في النظر إلى قاعدة البيانات على أنها مجموعة من الجداول (Tables) أو علاقات تسمى (Relations) ومن هنا جاءت تسمية النموذج وكل جدول يجب أن يكون له اسم (لا يوجد أكثر من جدول يحمل نفس الاسم).والعلاقة هي عبارة عن مصطلح رياضي وتمثل جدولاً ذا بعدين (صفوف وأعمدة)، ولا توجد هناك أهمية لترتيب الصفوف أو الأعمدة.حيث تمثل الصفوف مجموعة سجلات الجدول(Records or Tuple) وتمثل الأعمدة الصفات لهذا الجدول (Attributes) ويجب أن يكون لكل صفة مجال (Domain) من القيم التي يمكن أن يحتويها هذا العامود. وترتبط هذه الجداول مع بعضها بواسطة روابط..ويجب أن يكون لكل جدول مفتاح رئيس (Primary Key) لتمييز الصفوف عن بعضها والنقطة التي تمثل تقاطع الصف مع العمود (الصفة) تمثل قيمة لهذا الصف.و سنقوم في بقية أجزاء هذه الوحدة بتقديم وصفاً لقواعد البيانات العلائقية (Relational Database) من حيث مكوناتها وأهم خصائصها.

الجدول التالي يمثل معلومات الطالب (Student) في قاعدة بيانات إحدى الجامعات

- اسم الجدول Student
- كل صف يمثل معلومات تخص طالباً واحد فقط.
- المفتاح الرئيس للجدول هو St _No كل طالب يجب أن يكون له رقم مختلف عن بقية الطلاب.
- الصفة Dept_Code تمثل القسم الذي ينتمي إليه أي طالب .
- نقطة تقاطع الصفة (Gpa) العمود مع الصف الثالث تمثل المعدل التراكمي للطالب رقم ٢٠٠١ - ٠١ -١٠ .
- مجال القيم:كل صفة يجب أن يكون لها مجال ثابت من القيم فمثلا Gpa يجب أن تحتوي على رقم حقيقي بين ١.. ٥. القسم Dept_Code يجب أن يكون أحد الأقسام الدراسية الموجودة في الجامعة.



Student				
St_No	St_Name	Dept_Code	Birth_Date	Gpa
2000-01-101	Ali	Comp	12-08-1980	4.2
2001-02-99	Khalid	Math	10-10-1982	3.5
2001-01-10	Sami	Comp	01-01-1981	3.75

- لا توجد هناك أهمية لترتيب الصفوف أو الأعمدة.فمثلا يمكن أن يكون الجدول السابق على الشكل التالي:

Student				
St_No	St_Name	Gpa	Birth_Date	Dept_Code
٢٠٠١ -٠١- ١٠	Sami	٣,٧٥	٠١ -٠١- ١٩٨١	Comp
٢٠٠١ -٠٢- ٩٩	Khalid	٣,٥	١٠ -١٠- ١٩٨٢	Math
٢٠٠٠ -٠١- ١٠١	Ali	٤,٢	١٢ -٠٨- ١٩٨٠	Comp

مفاتيح الجداول (العلاقات):

تعتبر المفاتيح من أهم خصائص قواعد البيانات العلائقية حيث إنها تكون المميز لجدول معين من جهة والرابط الذي يربط الجداول المختلفة مع بعضها من جهة أخرى . ويمكن تقسيم المفاتيح في قواعد البيانات العلائقية إلى عدة أقسام :

أ - **المفتاح الأعظم** (Super Key): وهو أقل مجموعة من الصفات التي يمكن أن تميز الصف في الجدول عن بقية الصفوف الأخرى .فمثلا هذه المجموعة من الصفات يمكن أن تكون مفتاحا أعظم.

St_No
St_No, St _Name
St_No ,dept_code

ب -**المفتاح المرشح** (Candidate Key) : وهو الصفة (مجموعة الصفات) التي يمكن اختيارها كمفتاح رئيس للجدول ويجب أن يكون هناك أكثر من صف له نفس القيمة لهذه الصفة أو الصفات وكذلك يجب أن يكون له قيمة (ليس Null) .

ولكن كما لاحظنا فإن St_No, St _Name هي مفتاح أعظم ولكنه ليس مفتاحا مرشحا ليكون مفتاحا رئيسا لأن St_No وحدة يكفي لتمييز أي صف عن بقية الصفوف، ،لذلك فإن St_No يعتبر مفتاحا مرشحا ليكون مفتاحا رئيسيا .

ج - **المفتاح الرئيس** (Primary Key) : وهو المفتاح الذي تم اختاره من مجموعة المفاتيح المرشحة ليكون محددا لكل صف في الجدول .يمكن أن نختار St_No ليكون مفتاحاً رئيسا .

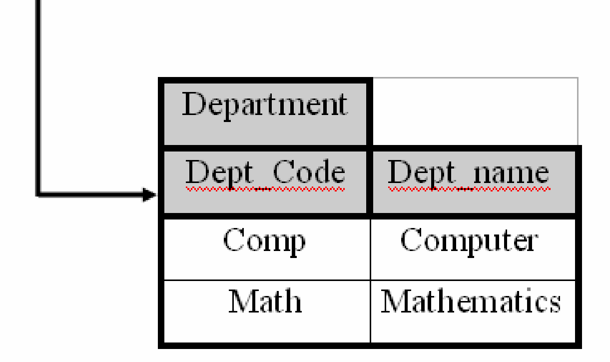
د - **المفتاح الثانوي** : هو عبارة عن صفة أوصفات تستخدم لغايات الاسترجاع ،فمثلا لو كان لدينا جدول يحتوي على قائمة بالعملاء فالمفتاح الرئيس هو رقم العميل Customer _id

ولكن إذا أردنا أن نسترجع رقم هاتف عميل معين (ولكن من سيحفظ أرقام العملاء ؟!) ففي هذه الحالة عادة ما يستخدم الاسم في عملية البحث وليس الرقم ،فيتم اختيار اسم العميل كمفتاح ثانوي .

Customer _id	Customer name	tel	Address

هـ - **المفتاح الأجنبي** (Foreign Key) : وهو صفة أو صفات تشير إلى مفتاح رئيس أو قيمة غير مكررة(Unique) في جدول آخر فمثلا تمل الصفة (Dept_Code) في جدول المتدرب (Student) مفتاح أجنبيا(Foreign Key) لجدول الأقسام (Department)

Student				
St _No	St _Name	Gpa	Birth_Date	Dept_Code
2001-01-10	Sami	3.75	01-01-1981	Comp
2001-02-99	Khalid	3.5	10-10-1982	Math
2000-01-101	Ali	4.2	12-08-1980	Comp

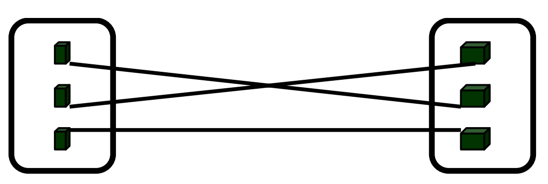


Department	
Dept_Code	Dept_name
Comp	Computer
Math	Mathematics

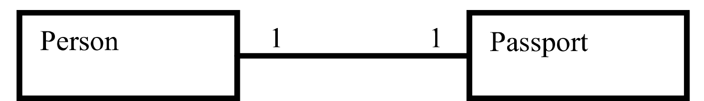
التشاركية بين الجداول (العلاقات):

وتمثل الدرجة التي ترتبط بها الجداول مع بعضها فيجب أن تحدد هذه الروابط بشكل واضح لمعرفة كيفية ارتباط هذه الجداول مع بعضها . هناك ثلاث درجات لارتباط الجداول :

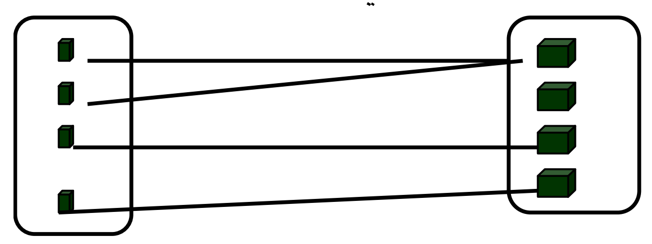
١. **واحد ــ واحد** (١:١): وهذا يعني أن قيمة واحدة في الجدول الأول تقابل قيمة واحدة فقط في الجدول الثاني



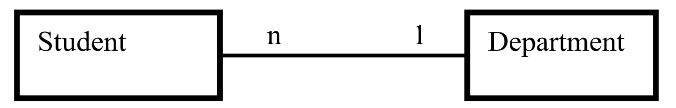
فمثلا يمكن أن نحدد على سبيل المثال أن لكل شخص جواز سفر واحد فقط وأن جواز السفر يعود لشخص واحد فقط .



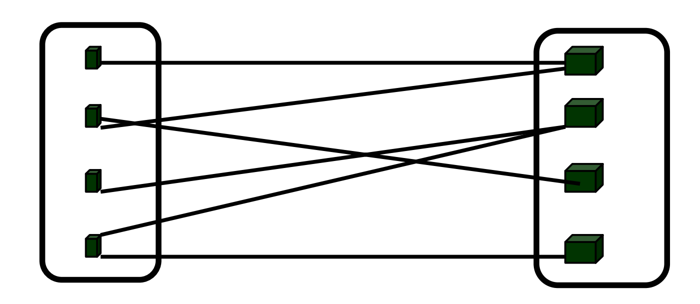
٢. **واحد ــ متعدد أو متعدد ــ واحد** (١: N أو N :١) وهـذا يعني أن قيمـة في الجدول الأول تقابل قيمـة في الجدول الثاني وأن القيمة في الجدول الثاني يمكن أن يقابلها قيمة أو أكثر في الجدول الأول.



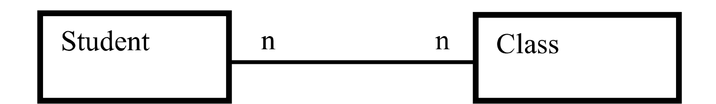
فمثلا يجب أن يتبع المتدرب لقسم واحد فقط وفي الوقت نفسه يمكن أن يكون هنالك أكثر من طالب ينتمي لهذا القسم .



٣. **متعدد ــ متعدد(N:N)** : وهذا يعني أن قيمة في الجدول الأول تقابل قيمة أو أكثر في الجدول الثانِي وأن القيمة في الجدول الثاني يمكن أن يقابلها قيمة أو أكثر في الجدول الأول.



فمثلا يمكن للطالب أن يسجل في أكثر من شعبة وكذلك الشعبة يمكن أن يسجل فيها أكثر من طالب.



تمارين

١ - عرف ما يلي :

- المفتاح المرشح
- المفتاح الرئيس
- المفتاح الأجنبي

٢ - وضح باستخدام الرسم علاقة الإشراف بين المدرس و الطالب.

٣ - وضح باستخدام الرسم نوع العلاقة بين القسم ومجموعة الاختصاصات في ذلك القسم في مستشفى.

٤ - أعط مثالا مع الرسم نوع التشاركية في قاعدة بيانات مستشفى

- واحد ـ واحد (N :N)
- واحد ـ متعدد (N:١)
- متعدد ـ متعدد (N :N)



تصميم قواعد البيانات

نموذج الكيانات والعلاقات

نموذج الكيانات والعلاقات



الجدارة:

معرفة عناصر نموذج العلاقات والكيانات والقدرة على تحويل ناتج عملية التحليل لنظام إلى نموذج العلاقات والكيانات.

الأهداف:

- أن يتعرف التدرب على عناصر نموذج العلاقات والكيانات .
- أن يحول المتدرب ناتج عملية التحليل للنظام إلى نموذج مفاهيم .

مستوى الأداء المطلوب:

تحويل ناتج عملية التحليل للنظام إلى نموذج مفاهيم نسبة ١٠٠٪.

الوقت المتوقع للتدريب:

٤ ساعات .

الوسائل المساعدة:

قلم + دفتر

متطلبات الجدارة:

أن يكون المتدرب قد أتقن الجدارة في الوحدات السابقة .

مقدمة :

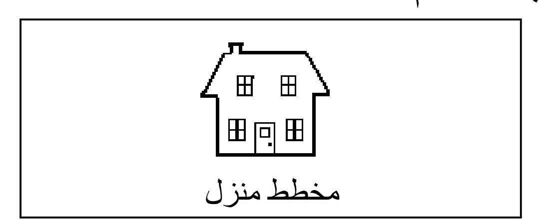
إن هدف عملية التصميم هو الوصول إلى فهم صحيح للنظام للمساعدة في عملية تطوير هذا النظام، وهذا ليس بالأمر السهل إذ لابد من وجود مقياس صحيح للحكم على هذا الفهم. ومن هنا برزت الأهمية لاستخدام العديد من الأدوات التي تساعد المصمم لوضع التصور والفهم الصحيحين لعمل هذا النظام.ومن هذه الأدوات استخدام النماذج التمثيلية التي تصف مكونات النظام وكيفية ارتباطها مع بعضها.وسنقوم في هذا الفصل بدراسة كيفية تمثيل البيانات باستخدام **نموذج الكيانات والعلاقات** Entity **Relationship (ER)Diagram**.

النماذج :

ما هو النموذج ؟

النموذج عبارة عن وصف رسومي (تمثيلي) لوصف الحقائق التي لا يمكن رؤيتها مباشرة.

وبعبارة أخرى هو وصف مجرد للكائنات الحقيقة. **نموذج البيانات** هو عبارة عن تمثيل بسيط لوصف تراكيب البيانات المعقدة في واقع الحياة العملية على شكل رسومي دون النظر إلى مكان وكيفية تخزين أو الوصول إلى هذه البيانات.ويستخدم هذا النموذج كوسيلة اتصال مابين المصمم من جهة وبين المبرمجين والمستخدمين من جهة أخرى. إذ حتى لو كان لدينا العديد من المبرمجين المحترفين فلا نستطيع الحصول على نظام جيد دون أن يكون هذا النظام قد صمم بشكل صحيح .والشكل التالي يبين مواصفات لمنزل وهذا الشكل يكون كوسيلة اتصال مابين الشخص الذي يرغب في بناء المنزل (الزبون) وكذلك بين المهندس(المصمم) من جهة وبين المقاول (المنفّذ) الذي سيقوم ببناء المنزل، وفي بناء أنظمة قواعد البيانات يمثل الزبون صاحب النظام ويمثل المصمم (مصمم قاعدة البيانات) والمقاول المنفذ هو مجموعة المبرمجين التي تقوم ببناء النظام .



مخطط منزل

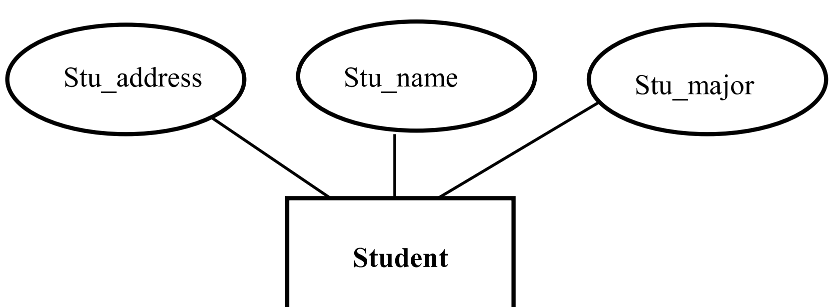
نموذج الكيانات والعلاقات:

هو عبارة عن نموذج لتمثيل كيانات النظام وصفاتها وكيفية ارتباط هذه الكيانات مع بعضها باستخدام رموز رسومية..ولنتعرف الآن على عناصر هذا النموذج:

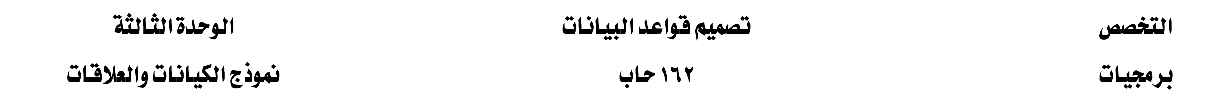
مجموعة الكيانات (**Entity Set**) وتمثل المجموعة التي تنتمي إليها مجموعة الكائنات(Objects) المتشابهة وتمثل بجدول في قاعدة البيانات العلائقية . و **الكيان**(Entity) هو عبارة عن كائن أو شيء محط الاهتمام في النظام وعلينا أن نقوم بجمع وتسجيل البيانات عن هذا الكيان. مثلا المتدرب ، المقرر، المدرس و الشعبة تعتبر كيانات مهمة في نظام قاعدة البيانات لجامعة .ويمثل الطبيب و المريض و وصفة العلاج كيانات مهمة في قاعدة بيانات لمستشفى . ويرمز لمجموعة الكيانات بمستطيل يحتوي على اسم الكيان .

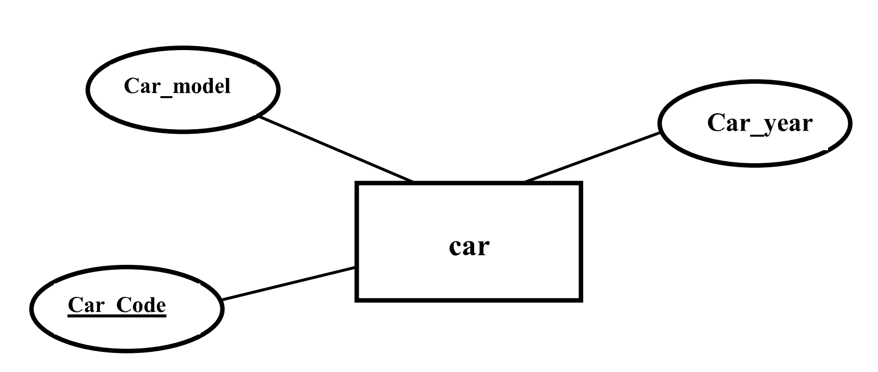


الخصائص أو الصفات (Attributes): هي عبارة عن الصفات المميزة للكيان، وبعبارة أخرى هي المعلومات الواجب تخزينها عن كائن معين وتمثل بأعمدة الجدول في قاعدة البيانات العلائقية..فمثلا لكل طالب يجب أن نسجل الاسم، الرقم، تاريخ الميلاد، التخصص، ولمنتج معين يكون الرقم الوصف، الطول، العرض، اللون.ويرمز للصفة بشكل بيضاوي يحتوي على اسم الصفة وتربط الصفة مع الكيان بواسطة خط مستقيم.



ولكل صفة يجب أن نحدد **مجال القيم (Domain)**: وهو مجموعة القيم لهذه الصفة فمثلا رقم المتدرب يجب أن يكون عدداً صحيحاً من عشر خانات، واسم المتدرب يجب أن يحتوي على قيم رمزية بطول ٣٠ حرف، والمعدل التراكمي يجب أن يحتوي على عدد كسري مابين ٠.. ٥ مثلا (٣,٥). تاريخ الميلاد يجب أن يكون مقبولاً بحيث لا يتجاوز عمر المتدرب عند القبول ٢٢ سنة. وبعض الصفات يمكن أن تشترك في نفس مجال القيم فمثلا القسم الدراسي للطالب والمدرس يكون اسماً من أسماء الأقسام في الجامعة. والصفة (مجموعة الصفات) التي تم اختيارها كمفتاح رئيس (**primary key**) تُمثل كأي صفة ولكن يوضع خط تحت الاسم.

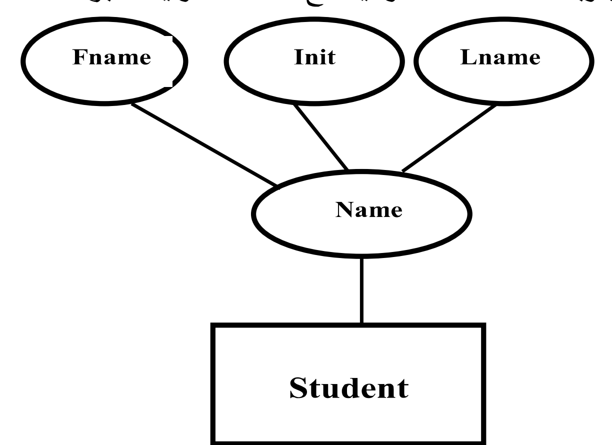




وفي عملية تحديد الصفات للكيانات لابد من أن نحدد

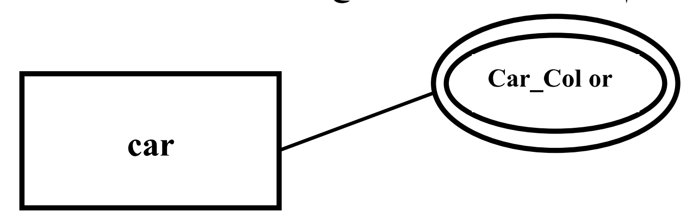
أ - الصفات البسطة والمركبة **Simple and Composite Attributes** :

وتقسم إلى صفات بسيطة أي لا يمكن تجزئتها مثل رقم الطالب، الجنس تاريخ الميلاد. أو مركبة أي يمكن تجزئتها كالاسم (الاسم الأول، الثاني، واسم العائلة)، العنوان (المدينة، الحي، الشارع، رقم المنزل). .ويرمز للصفة المركبة بشكل بيضاوي ترتبط معه أشكال بيضاوية أخرى يحتوي كل منها على اسم الصفة الفرعية وتربط الصفات الفرعية مع الصفة الرئيسة بواسطة خط مستقيم .



ب -صفات وحيدة أو متعددة القيم **Single-Valued or Multiple-Valued Attributes** :

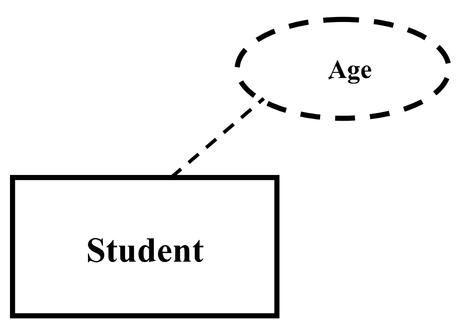
الصفات التي تحتوي على قيمة واحدة مثل (رقم السيارة، تاريخ الصنع) أو عدة قيم مثل لون السيارة (فيمكن أن يكون هناك لون للسقف، الجسم، الجوانب) وكذلك يمكن أن يكون للمدرس أكثر من رقم هاتف أو أكثر من بريد إلكتروني. ويرمز للصفة متعددة القيم بشكل بيضاوي داخل شكل بيضاوي آخر يحتوي على اسم الصفة وتربط الصفة مع الكيان بواسطة خط مستقيم.



ج ـ الصفات المشتقة (Derived Attributes):

وهي الصفات التي يمكن اشتقاقها من صفات أخرى ويرمز لها بشكل بيضاوي متقطع يحتوي على اسم الصفة وترتبط مع الكيان بخط مستقيم متقطع أيضا كما في الشكل التالي. مثل عُمر المتدرب يمكن حسابه على أنه الفرق بين تاريخ الميلاد والتاريخ الحالي.

العمر = التاريخ الحالي – تاريخ الميلاد.

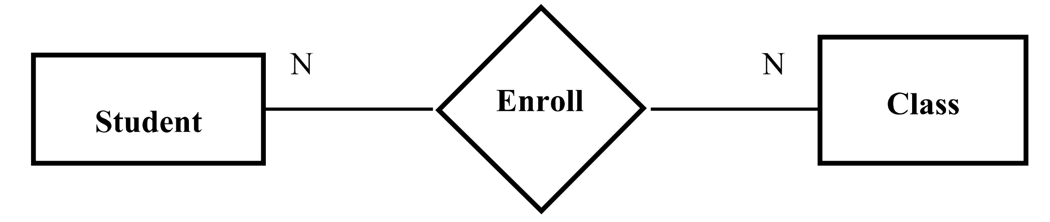


الصفات المشتقة يجب أن لا تخزن ولكن توضع طريقة لحسابها عند عملية الاسترجاع.ولكن قد نخزن بعض الصفات المشتقة إذا كانت عملية حسابها تأخذ وقتا كبيرا وفي نفس الوقت يتم طلبها بشكل كبير مثل المعدل التراكمي للطالب.

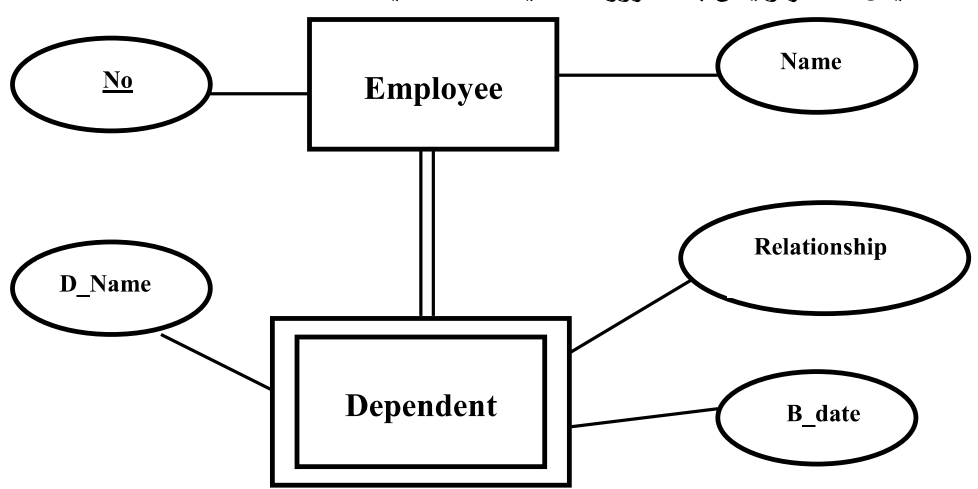
الروابط أو العلاقات (Relationships):

وهي عبارة عن الرابط أو العلاقة مابين الكيانات واسم هذه الرابطة يجب أن يعبر عن كيفية هذا الترابط ويكون على شكل فعل (ينتمي، يحتوي، يسجل، يتكون من....). ويرمز لها بشكل معين يحتوي على اسم الرابط أو العلاقة.وكذلك لكل علاقة درجة تشاركية. وتبين مقدار التشارك مابين الكيانات إما واحد ــ واحد (١:١) أو واحد ــ متعدد(**N :1**) أو متعدد ـ متعدد (**N:N**).

فالطالب يسجل في شعبة أو أكثر والشعبة يسجل فيها مجموعة من الطلاب .



الكيانات الضعيفة: وهي عبارة عن الكيانات التي لا توجد مستقلة بنفسها في النظام وبعبارة أخرى فإن وجودها يعتمد على وجود كيان آخر فمثلا لنفرض أن مؤسسة ما تسجل معلومات عن أسماء الأشخاص التابعين للموظف مثل الأبناء، الزوجة أو الوالدين. فوجود معلومات التابع مرتبط بوجود الموظف وفي هذه الحالة يختار المفتاح الرئيس للكيان الرئيس مع صفة من صفات التابع (مثل الاسم) لتشكل مفتاحا رئيسا للكيان التابع ويوضع تحته خط مقطع. ويرمز للكيان الضعيف بمستطيل داخل مستطيل يحتوي على اسم الكيان الضعيف ويرتبط مع الكيان الرئيس بخطين مستقيمين (يعني أن وجود الكيان الأول شرط لوجود الكيان الآخر وليس بالضرورة للكيانات الضعيفة فقط).

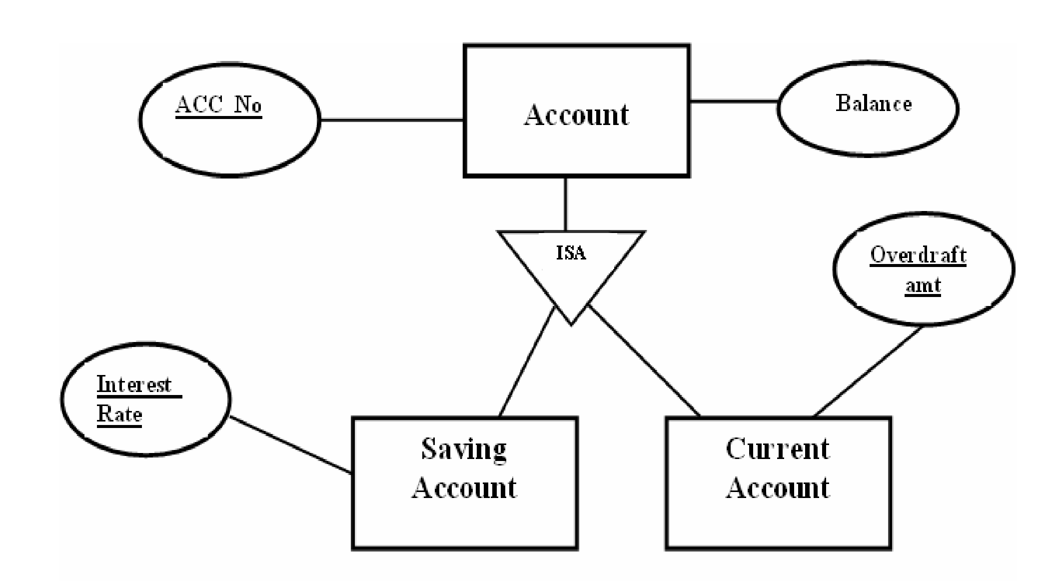


تمثيل الأنواع الرئيسة والأنواع الفرعية (Supertype and Subtype) :

هناك بعض الكيانات الفرعية التي تتبع إلى نوع رئيس (أعلى) Supertype فمثلا بالنسبة للحساب البنكي يمكن أن يكون هناك أكثر من نوع للحسابات ولكن جميع هذه الحسابات تشترك في الكثير من الصفات ففي هذه الحالة نقوم بإنشاء كيان الحساب البنكي Account بحيث يحتوي على جميع هذه الصفات ،ثم بعد ذلك نقوم بإنشاء كيانات فرعية للحسابات يحتوي كل منها على الصفات الخاصة بهذا النوع فقط.

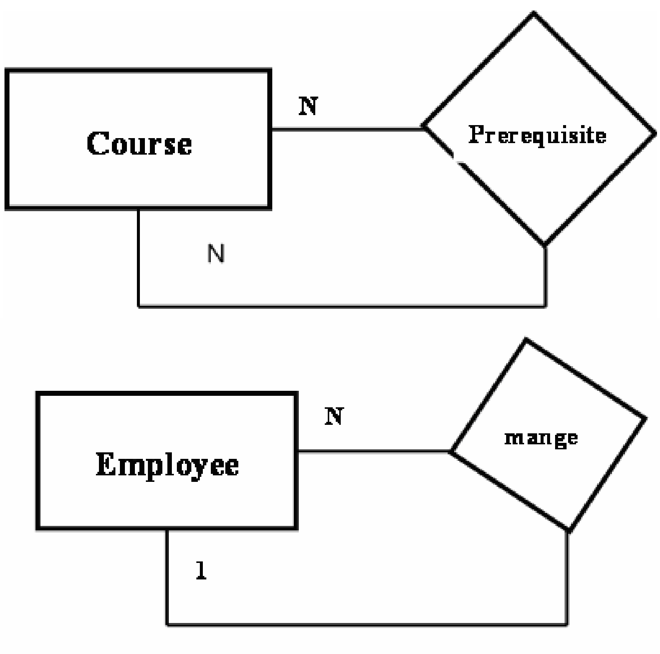
مثال: لنفرض أن لكل الحساب حقل يمثل رقم الحساب وحقل يمثل الرصيد الحالي وفي نفس الوقت لدينا نوعين من الحسابات: الحساب الجاري (Current Account) وفيه الصفة (Overdraft Amount) وهي أعلى قيمة يسمح لصاحب الحساب أن يسحبها عندما لا يكون لديه رصيد. والنوع الثاني حساب التوفير وفية صفة معدل الفائدة(Interest Rate) .

وتمثل العلاقة بين الأنواع الرئيسة العليا والأنواع الفرعية بمثلث مقلوب يحتوي على (**ISA**) بمعنى يكون.



تمثيل علاقة الكيان مع نفسه (Recursive):

وفي هذه الحالة نبين كيفية تمثيل ارتباط الكيان مع نفسه، فمثلا نفرض أن المقرر الدراسي يمكن أن يكون لديه متطلب سابق أو أكثر (وهذا المتطلب هو عبارة عن مقرر) وكذلك يجب أن يكون للموظف مدير واحد فقط (والمدير بدوره هو أيضا موظف)



حالة دراسية: سنقوم في هذا المثال بعملية تحويل عملية تحليل شركة ما إلى نموذج مفاهيم (نموذج الكيانات و العلاقات ER Diagram). حيث إن الشركة تهتم بتسجيل معلومات عن الأقسام والمشاريع التي تنفذها الشركة وكذلك عن الموظفين العاملين فيها والتابعين لهؤلاء الموظفين .

١ - تقسم الشركة إلى عدة أقسام ولكل قسم اسم واحد ورقم (لا يجوز أن يكون هناك أكثر من قسم بنفس الاسم أو الرقم)، لكل قسم موظف يدير هذا القسم ويجب معرفة التاريخ الذي بدأ فيه هذا الموظف بإدارة القسم ،ولكل قسم موقع واحد فقط.

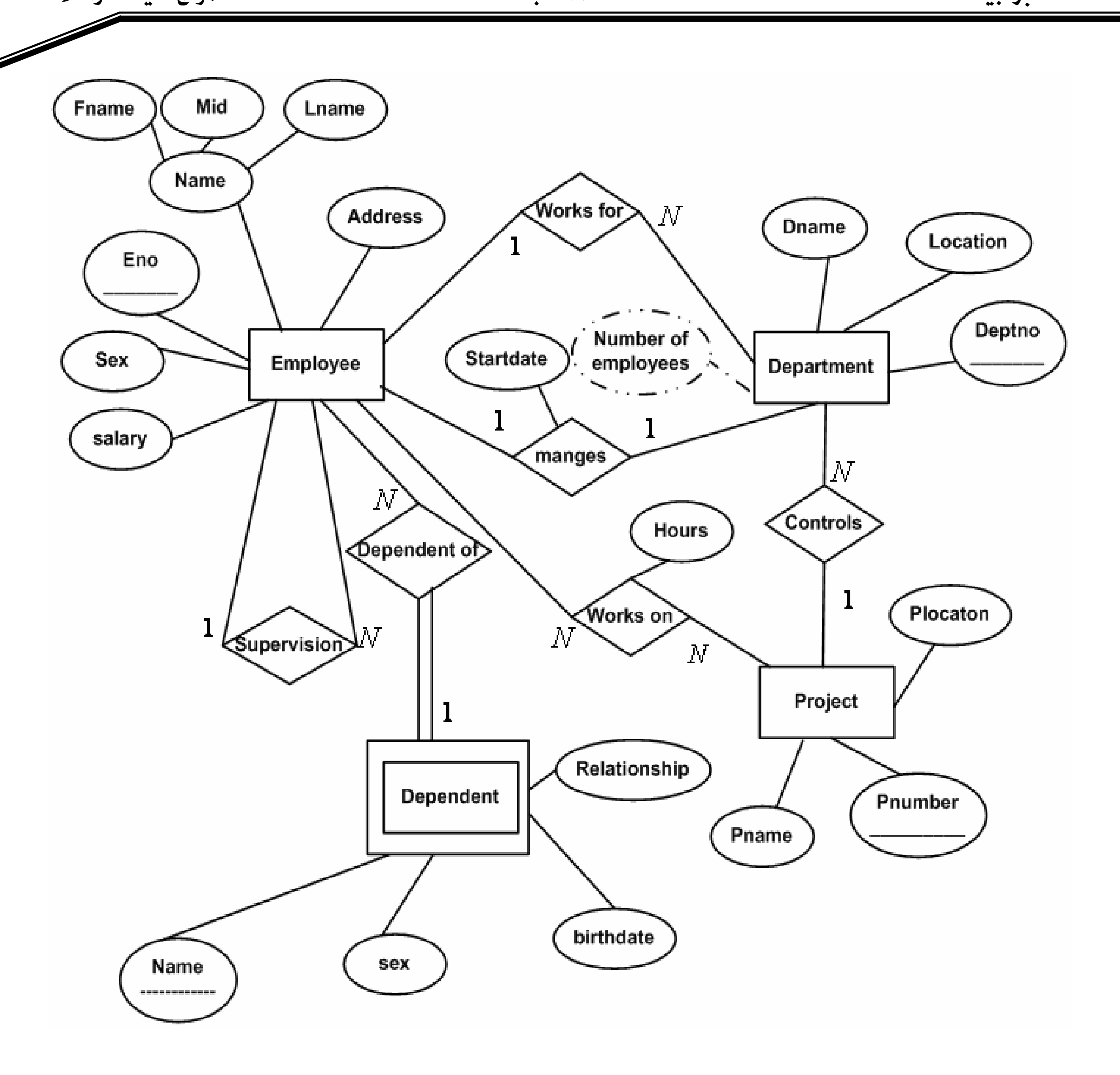
٢ - القسم يمكن أن يدير عدة مشاريع ولكل مشروع رقم واسم ومكان تنفيذ.

٣ - يمكن أن يعمل في القسم موظف أو أكثر ولكن الموظف يجب أن يتبع لقسم واحد فقط ونحتفظ بالمعلومات التالية عن الموظف (الرقم لكل موظف رقم يميزه عن بقية الموظفين، الاسم (الاسم الأول، الاسم الثاني واسم العائلة)، الجنس ، الراتب والعنوان.

٤ - الموظف يمكن أن يعمل في عدة مشاريع (وليس بالضرورة أن يدار المشروع من نفس القسم الذي يتبع إليه الموظف) ونحتفظ بعدد الساعات التي عملها الموظف في كل مشروع.

٥ - تحتفظ الشركة بمعلومات عن التابعين لكل موظف مثل الاسم ،تاريخ ، الميلاد، الجنس، صلة القرابة .

٦ - تهتم الشركة بمعرفة عدد الموظفين الذين يتبعون لقسم معين.


Fname
Mid
Lname
Name
Address
Eno
Sex
salary
Employee
Works for
1
N
Dname
Location
Deptno
Department
Startdate
Number of employees
1
manges
1
N
Controls
N
Dependent of
Hours
1
1
Supervision
N
Works on
N
N
Plocaton
Project
1
Dependent
Relationship
Pname
Pnumber
Name
sex
birthdate

تمارين

١ -عرف ما يلي:

- الكيان Entity .
- الصفة Attribute .
- العلاقة Relationship .

٢ -اذكر أنواع التشاركية بين الكيانات مع ذكر الأمثلة واستخدام الرسم .

٣ - ما الفرق بين الصفات وحيدة القيم والصفات متعددة القيم؟ وكيف تُمثل باستخدام الرسم ؟

٤ -اذكر مثالاً على الكيانات الضعيفة وارسم نموذج مفاهيم لتوضيح ذلك .

٥ - ارسم نموذج العلاقات والكيانات لكل مما يلي :

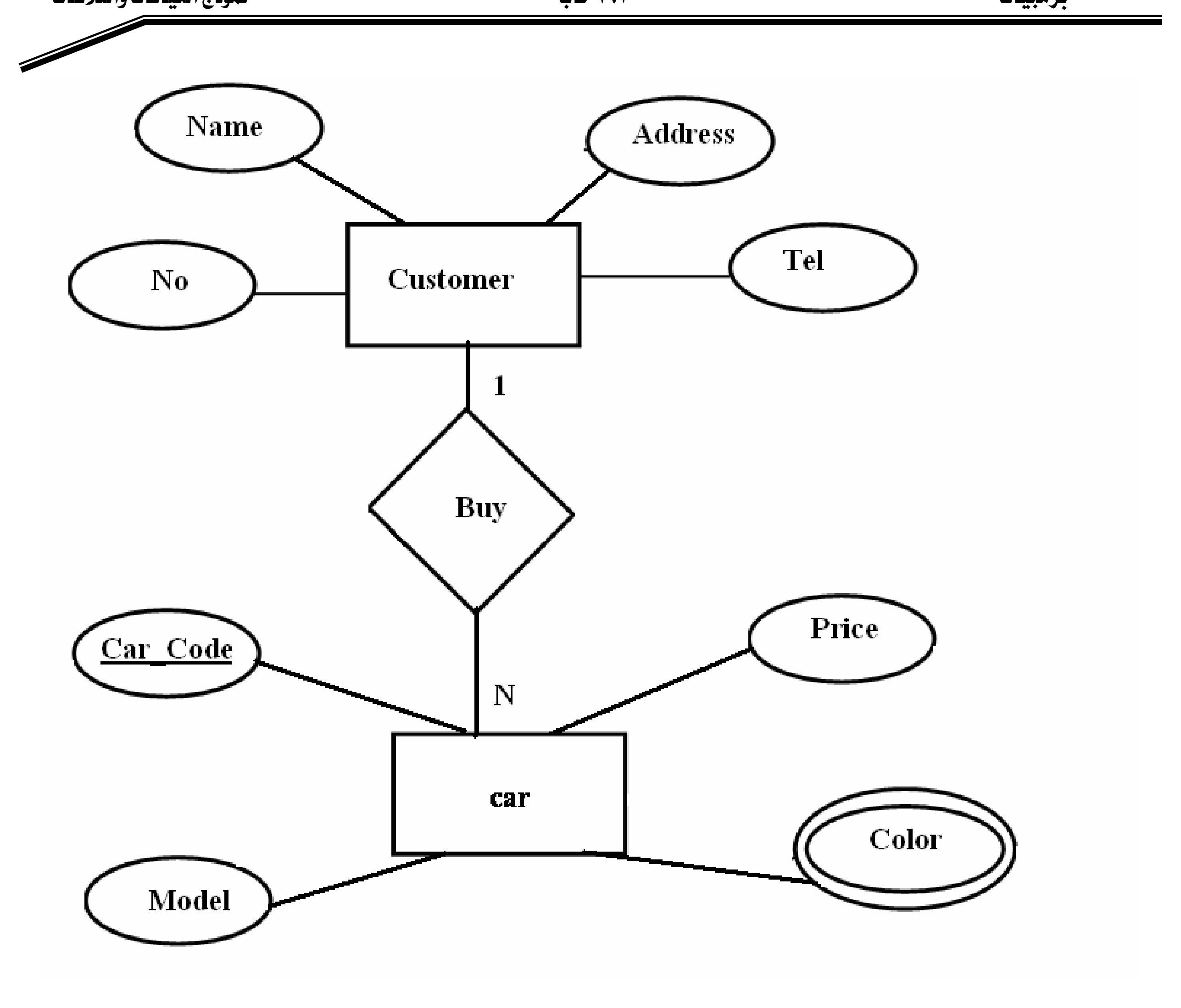
أ - في قاعدة بيانات لمكتبة المؤلف يمكن أن يؤلف أكثر من كتاب والكتاب يمكن أن يشترك في تأليفه أكثر من مؤلف ،وكذلك يجب أن يتبع الكتاب لموضوع واحد فقط.

ب -في قاعدة بيانات لمستشفى يمكن أن يشرف الطبيب على أكثر من مريض والمريض يجب أن يشرف عليه طبيب واحد .

ج - في قاعدة بيانات لجامعة يمكن أن يدرس المدرس أكثر من شعبة والشعبة تكون لمقرر واحد فقط ويجب أن يدرسها مدرس واحد .

٦ - ارسم نموذج العلاقات والكيانات كاملاً لنظام مبيعات بحيث يمكن للزبون أن يشتري أكثر من منتج ويجب أن تتم عملية الشراء من خلال فاتورة والفاتورة تحرر من قبل موظف واحد فقط وكذلك فإن لكل منتج كمية معينة داخل المحل وكمية أخرى في المستودع وفي حال نفاذ الكمية من المحل يقوم بطلب كمية أخرى من المستودع والمستودع بدورة يقوم بتزويد المحل بمنتج أو أكثر في نفس الوقت.

٧ - صف على شكل نقاط نموذج العلاقات والكيانات التالي :


Name
Address
No
Customer
Tel
1
Buy
Car_Code
Price
N
car
Color
Model



تصميم قواعد البيانات

الصيغ المعيارية

الصيغ المعيارية

٤

الجدارة:

القدرة على تحويل الجداول إلى الصيغة المعيارية الثالثة **3NF** .

الأهداف:

- أن يتعرف المتدرب على مشاكل تكرار البيانات(Data Anomalies) :
- أن يتعرف المتدرب على الاعتمادية الوظيفية
- أن يستطيع المتدرب تعريف الصيغة المعيارية الأولى
- أن يستطيع المتدرب تعريف الصيغة المعيارية الثانية
- أن يستطيع المتدرب تعريف الصيغة المعيارية الثالثة

مستوى الأداء المطلوب:

أن يتقن المتدرب عملية تحويل الجداول إلى الصيغة المعيارية الثالثة **3NF** بنسبة١٠٠٪.

الوقت المتوقع للتدريب:

ساعتان

الوسائل المساعدة:

قلم + دفتر

متطلبات الجدارة:

أن يكون المتدرب قد أتقن الجدارة في الوحدات السابقة .

مقدمة :

إن عملية وضع تصميم قاعدة البيانات في الصيغة المعيارية يشكل لبنة أساسية في عملية التصميم الجيد لقاعدة البيانات.وتتم هذه العملية على عدة مراحل يتم خلالها تخليص قاعدة البيانات من التكرار غير المسوغ للبيانات بالاعتماد على قوانين الاستنتاج والاعتمادية الوظيفية. و.سنقوم في هذا الفصل بالتعرف على الشروط و القوانين اللازمة للوصول بقاعدة البيانات إلى المستوى المعياري الثالث(Third Normal Form **3NF**).

مشاكل تكرار البيانات (Data Anomalies) :

Employee_department						
Empno	Ename	Job	Salary	Deptno	Dname	Loc
101	Sami	clerk	3000	10	Accounting	Riyadh
205	Khalid	manager	2500	10	Accounting	Riyadh
303	Ali	salesman	1200	20	Sales	Jeddah
502	Saeed	salesman	2100	20	Sales	Jeddah
601	Salem	clerk	1000	30	Operation	Dmmam

نلاحظ في الجدول السابق أن معلومات الموظف والقسم الذي يعمل فيه موجودة في جدول واحد ونتيجة ذلك تكرار بعض البيانات مثل اسم وموقع القسم في كل سجل وهذا يؤدي إلى عدة مشاكل :

١ - **مشكلة الإضافة** :أي إننا لا نستطيع أن نظيف قسماً جديداً إلا إذا كان القسم يحتوي على، موظف ،لأن المفتاح الرئيس للجدول هو رقم الموظف.

٢ - **مشكلة التعديل** : نلاحظ تكرار اسم وموقع القسم فإذا قمنا بتعديل موقع (Loc)القسم رقم ٢٠ من Jeddah إلى Riyadh فلابد من إجراء عملية التعديل لجميع الموظفين في هذا القسم وإلا ستؤدي هذه العملية إلى عدم توافقية البيانات أي نفس رقم القسم ولكن أكثر من موقع . وكذلك إذا تمت عملية التغيير عند الموظف رقم ٣٠٣ عن طريق الخطأ . وبالتالي لو قمنا بعملية استرجاع لجميع الموظفين الذين يعملون في Jeddah فإن الموظف رقم ٣٠٣ لن يظهر بين الموظفين .

٣ - **مشكلة الحذف** : نلاحظ أن القسم رقم ٣٠ يحتوي على موظف واحد فقط ،فلو قمنا بحذف الموظف رقم ٦٠٦ فإن معلومات القسم رقم ٣٠ سوف تختفي من الجدول .

الاعتمادية الوظيفية (Functional Dependency FD):

وهي اعتماد قيمة إحدى صفات الكيان على قيمة صفة(صفات) أخرى ويرمز لها بالرمز (⟵)

مثال $A \longrightarrow B$

يعني أن B تعتمد اعتمادا وظيفيا على A وهنا نستطيع أن نقول أن قيمة A تحدد قيمة B . ومن خلال تحديد الاعتمادية نستطيع أن نحدد المكان الذي يجب أن توضع فيه الصفة وهذا بالتالي يؤدي إلى وضع البيانات في المكان الصحيح ونتخلص من عملية تكرار البيانات وما يترتب على التكرار من مشاكل(Anomalies) .

مثال: لكل موظف اسم واحد فقط ولكل موظف قسم واحد يعمل فيه إذا :

FD1 : Empno ⟶ Ename

FD2 : Empno ⟶ Deptno

ويمكن أن نعيد كتابة هذه الاعتمادية على الشكل التالي

FD1 : Empno ⟶ Ename,Deptno

FD :Functional Dependency

قواعد الاستنتاج

وهي عبارة عن مجموعة من القواعد تستخدم في عملية تحديد الاعتمادية الوظيفية (Functional Dependency FD) وتتلخص هذه القواعد بستة قواعد على النحو التالي :

١ - الانعكاسية **Reflexive** :إذا كانت Y جزء من X ((Y محتواه في X)) فإن X تحدد Y

$1\text{-}\ X \supseteq Y :\quad X \rightarrow Y$

٢ - قاعدة الزيادة أو الإضافة **Augmentation** :إذا كان X تحدد Y فإن إضافة Z إلى X تعني أنه بالإمكان إضافة Z إلى Y

$2\text{-}\ \{X \rightarrow Y\} \models XZ \rightarrow YZ$

٣ - قاعدة التعدي **Transitive** : تعني أنه إذا كانت X تحدد Y وكانت Y تحدد Z فإن X تحدد Z .

$3\text{-}\ \{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z$

٤ - قاعدة الاتحاد **Union** : تعني أنه إذا كانت X تحدد Y و X تحدد Z فإننا نستطيع أن نقول أن X تحدد YZ .

4- $\{X\rightarrow Y, X\rightarrow Z\} \models X \rightarrow YZ$

٥ - قاعدة التقسيم Decomposition وهي عكس قاعدة الاتحاد

5- $\{X\rightarrow YZ\} \models X \rightarrow Y, X\rightarrow Z$

٦ - قاعدة التعدي الزائف **Pseudotransitive** تشبه قاعدة التعدي

6- $\{X\rightarrow Y, WY\rightarrow Z\} \models WX \rightarrow Y$

$\models$ تعني أنه إذا تحقق الطرف الأيسر فإننا نستطيع استنتاج الطرف الأيمن .

تعريف الصيغة المعيارية الأولى(First Normal Form 1NF):

نستطيع أن نقول أن الجدول في الصيغة المعيارية الأولى إذا كانت جميع أعمدة الجدول تحتوي على بيانات بسيطة أو مفردة(غير مركبة) أي إن كل عمود يحتوي على قيمة واحدة فقط .

مثال ١ يمثل الجدول التالي معلومات موظف Employee:

No	Name			Adresse		
	Fname	Mid	Lname	city	Street	House no
100	Ali	Salem	musa	Riyadh	Immam saud	210
120	Saeed	Eisa	Ali	Riyadh	King Fahad	202

نلاحظ في الجدول أن الاسم يتكون من ثلاثة أجزاء وكذلك العنوان فبالتالي لا نستطيع أن نخزن قيمة واحدة في عمود الاسم وكذلك بالنسبة للعنوان وهذا يخالف شروط قاعدة البيانات بأن العمود يجب أن يحتوي على قيمة واحدة فقط.وهذا يعني أن الجدول السابق لا ينطبق عليه شرط الصيغة المعيارية الأولى **1NF,** ولوضع الجدول في الصيغة المعيارية الأول يجب تقسيم الأعمدة المركبة إلى أعمدة بسيطة

No	Fname	Mid	Lname	city	Street	House no
100	Ali	Salem	musa	Riyadh	Immam saud	210
120	Saeed	Eisa	Ali	Riyadh	King Fahad	202

لقد قمنا بتقسيم الأعمدة المركبة إلى أعمدة بسيطة وبالتالي نستطيع أن نقول أن الجدول الآن في الصيغة المعيارية الأولى **1NF** .

مثال ٢ : يمثل الجدول التالي سجل ساعات العمل HOURS لموظف في عدد من المشاريع PROJECTS والقسم الذي يشرف على تنفيذ المشروع

NO	Name	Project_Code	Hours	Deptno	Dname
210	Ali	P1,p2,p3	12,20,40	10,20,20	Research, Operation, Operation
201	Salem	P1,p3	30,15	10,20	Research Operation
305	Ali	P2,p3	40,20	20,20	Operation, Operation

كما هو مبين في الجدول السابق فإن هناك عدداً من الأعمدة تحتوي على أكثر من قيمة مثل رمز المشروع Project_Code وكذلك عدد ساعات العمل Hours والأقسام Deptno التي تشرف على المشاريع . وهذا يعني أن الجدول ليس في الصيغة المعيارية الأولى، ولتحويله يجب أن نقوم بتقسيم الجدول على النحو التالي للتخلص من هذه المشكلة.

NO	Name	Project_Code	Hours	Deptno	Dname
210	Ali	P1	12	10	Research
210	Ali	p2	20	20	Operation
210	Ali	p3	40	20	Operation
201	salem	P1	30	10	Research
201	salem	p3	15	20	Operation
305	Ali	P2	40	20	Operation
305	Ali	p3	20	20	Operation

ولكن تبرز هنا لدينا مشكلة جديدة وهي إيجاد مفتاح رئيسي للجدول إذ أصبح رقم الموظف لا يصلح لأن يكون مفتاحا رئيسا للجدول (Primary Key) لأن من شروط المفتاح الرئيس أن لا يتكرر في أكثر من صف .لنقوم الآن باستخدام الاعتمادية الوظيفية لمحاولة إيجاد المفتاح الرئيس للجدول

FD 1 :No → Name

حيث إن لكل موظف اسم واحد .

FD 2 : Project_Code→ Deptno

حيث إن لكل مشروع قسم واحد يشرف عليه .

FD 3 : Deptno →Dname

حيث إن لكل قسم اسم واحد.

أما بالنسبة لبقية العناصر فمثلا اسم الموظف لا يحدد شيئاً لأنه يوجد هناك أكثر من موظف اسمه Ali فالاسم لا يحدد الرقم وكذلك فإن علي يعمل في أكثر من مشروع .

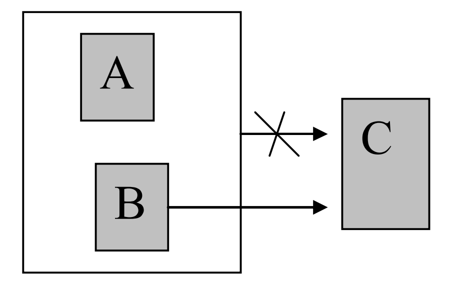
وكذلك رمز لمشروع لا يحدد عدد الساعات ولاالموظفين الذين يعملون فيه فالمشرع P1 يعمل فيه أكثر من موظف وبساعات مختلفة .
أما بالنسبة للقسم فلا يحدد الموظفين ولا المشاريع فمثلا القسم ٢٠ يشرف على أكثر من مشروع هذه المشاريع يعمل فيها أكثر من موظف .
ففي هذه الحالة يجب علينا القيام بمحاولة جديدة لإيجاد المفتاح الرئيس وتتلخص هذه العملية بمحاولة إيجاد مفتاح مركب(تركيب أكثر من صفة لتشكل المفتاح الرئيس) يقوم بتحديد جميع الصفات في الجدول :
سنقوم بأخذ رقم الموظف مع رقم المشروع

FD 4 :No, Project_Code→ name
FD 5 :No, Project_Code→ Deptno
FD 6 :No, Project_Code→ Hours
FD 7 : Deptno →Dname
FD 8 :No, Project_Code→ Name ,Hours, Deptno, Dname

FD4,FD5 تنطبق من FD1,FD2 حيث إن رقم الموظف وحدة يحدد الاسم وكذلك رمز المشروع يحدد القسم ، أما بالنسبة ل FD5 فإنها تنطبق لأن رقم الموظف ورمز المشروع يحددان عمل الموظف في ذلك المشروع ، وبالتالي نكون قد حصلنا على مفتاح رئيس لهذا الجدول وكذلك قمنا بوضعه في الصيغة المعيارية الأولى (**1NF**).

تعريف الصيغة المعيارية الثانية(Second Normal Form 2NF):
نستطيع أن نقول أن الجدول في الصيغة المعيارية الثانية:
١ - إذا كان الجدول في الصيغة المعيارية الأولى.
٢ - إذا لم يحتوي الجدول على اعتمادية جزئية.

الاعتمادية الجزئية: هي أن تعتمد بعض الأعمدة (الصفات) اعتمادا وظيفيا على جزء من المفتاح الرئيس



نلاحظ أن A,B تحدد C أي إن C تعتمد اعتمادا وظيفيا على A,B وكذلك أن B تحدد C أي إن C تعتمد اعتمادا وظيفيا B. وفي هذه الحالة نستطيع أن نقول أن هذا الجدول يحتوي على اعتمادية جزئية .

NO	Name	Project_Code	Hours	Deptno	Dname
210	Ali	P1	12	10	Research
210	Ali	p2	20	20	Operation
210	Ali	p3	40	20	Operation
201	Salem	P1	30	10	Research
201	Salem	p3	15	20	Operation
305	Ali	P2	40	20	Operation
305	Ali	p3	20	20	Operation

والآن هل الجدول السابق في الصيغة المعيارية الثانية ؟

وللإجابة على ذلك نجيب على السؤالين التاليين :

١ - هل الجدول في الصيغة المعيارية الأولى ؟

نعم، لأنه لا توجد هناك قيم متكررة ،كل عمود يحتوي على قيمة واحدة فقط .

٢ - هل توجد هناك اعتمادية جزئية ؟

ولمعرفة ذلك يجب أن نحدد الاعتمادية الوظيفية

FD 1 :No → Name

FD 2 : Project_Code→ Deptno,Dname

FD 3 :No, Project_Code→ name ,deptno, hours

المفتاح الرئيس هو No, Project_Code ولكن No يحدد Name إذا هناك اعتمادية جزئية وكذلك

Project_Code يحدد deptno وDname و هذه اعتمادية جزئية أخرى . وللتخلص من هذه المشكلة يجب أن نقوم بتقسيم الجدول إلى جداول بحيث يضم كل منها الجزء من المفتاح والأعمدة التي تعتمد عليه ونبقي فقط المفتاح المركب مع الأعمدة التي تعتمد عليه:

١ - نقوم بنقل اسم ورقم الموظف إلى جدول جديد ونبقي نسخة من رقم الموظف في الجدول الأصلي (لأنه جزء من المفتاح الرئيس) .

٢ - نقوم بنقل رمز المشروع ورقم القسم إلى جدول جديد ونبقي نسخة رمز المشروع في الجدول الأصلي (لأنه جزء من المفتاح الرئيس) .

٣ - نبقي بقية الأعمدة كما هي (عدد الساعات).

٤ - وبالتالي تصبح الجداول على النحو التالي بعد عملية التقسيم :

NO	Project_Code	Hours
210	P1	12
210	p2	20
210	p3	40
201	P1	30
201	p3	15
305	P2	40
305	p3	20

NO	Name
210	Ali
210	Ali
210	Ali
201	salem
201	salem
305	Ali
305	Ali

Project_Code	Deptno	Dname
P1	10	Research
p2	20	Operation
p3	20	Operation

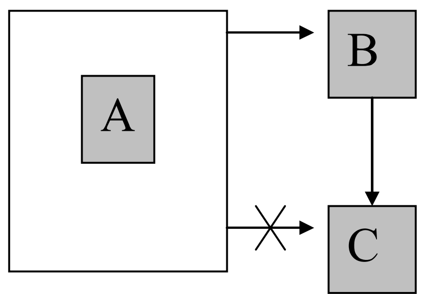
تعريف الصيغة المعيارية الثالثة (Third Normal Form 3NF):

نستطيع أن نقول أن الجدول في الصيغة المعيارية الثالثة:

١ - إذا كان الجدول فيالصيغة المعيارية الثانية.

٢ - إذا لم يحتوي الجدول على اعتمادية متعدية.

الاعتمادية المتعدية: هي أن تعتمد بعض الأعمدة (الصفات) اعتمادا وظيفيا على صفة غير المفتاح الرئيس.



نلاحظ أن A تحدد B, C أي إن B, C تعتمد اعتمادا وظيفيا على A وكذلك أن B تحدد C أي إن C تعتمد اعتمادا وظيفيا B. وفي حالة نستطيع أن نقول أن هذا الجدول يحتوي على اعتمادية متعدية.

والآن هل الجداول السابقة في الصيغة المعيارية الثالثة ؟

وللإجابة على ذلك نجيب على السؤالين التالين:

١ - هل الجداول في الصيغة المعيارية الثانية ؟

نلاحظ أن جميع الجداول في الصيغة المعيارية الثانية حيث لا يوجد فيها اعتمادية جزئية .

٢ - هل توجد هناك اعتمادية متعدية ؟

ولمعرفة ذلك يجب أن نحدد الاعتمادية الوظيفية لكل جدول

أ - الجدول الأول

FD 1 :No → Name

لا توجد اعتمادية متعدية .

ب - الجدول الثاني

FD 1 :No, Project_Code→ hours

لا توجد اعتمادية متعدية.

ج - الجدول الثالث

FD 1 : Project_Code→ Deptno,Dname

FD 2 : Deptno → Dname

المفتاح الرئيس هو Project_Code يحدد Dname وDeptno وفي نفس الوقت فإن وDeptno يحدد Dname أي إن هناك اعتمادية متعدية . وللتخلص من هذه المشكلة يجب أن نقوم بتقسيم الجدول إلى جداول بحيث يضم كل منها الأعمدة التي تعتمد على بعض ونبقي المفتاح مع الأعمدة التي تعتمد عليه وحدة فقط مع إبقاء المحدد الجديد (Deptno)

١ - نقوم بنقل رقم و اسم القسم إلى جدول جديد ونبقي نسخة من رقم القسم في الجدول الأصلي.

٢ - وبالتالي تصبح الجداول على النحو التالي بعد عملية التقسيم :

NO	Project_Code	Hours
210	P1	12
210	p2	20
210	p3	40
201	P1	30
201	p3	15
305	P2	40
305	p3	20

NO	Name
210	Ali
210	Ali
210	Ali
201	Salem
201	Salem
305	Ali
305	Ali

Project_Code	Deptno

Deptno	Dname

P1	10
p2	20
p3	20

10	Research
20	Operation

الآن نستطيع أن نقول أن هذه الجداول هي في الصيغة المعيارية الثالثة **3NF** وتعتبر هذه الصيغة مقبولة لمعظم مصممي قواعد البيانات .

تمارين

١. وضح المقصود بمشاكل تكرار البيانات(Data Anomalies) مع الأمثلة .

٢. ما هي الاعتمادية الوظيفية (Functional Dependency FD)؟

٣. اذكر قواعد الاستنتاج مع ذكر مثال من قاعدة بيانات تسجيل الطلاب على كل قاعدة .

٤. متى يكون الجدول في

أ - الصيغة المعيارية الأولى.

ب - الصيغة المعيارية الثانية .

ج - الصيغة المعيارية الثالثة .

٥. أعط مثالاً على كل مما يلي :

أ - جدول ليس في الصيغة المعيارية الأولى.

ب - جدول ليس في الصيغة المعيارية الثانية .

ج - جدول ليس في الصيغة المعيارية الثالثة .

٦. هل الجدول التالي في الصيغة المعيارية الثالثة ؟ إذا لم يكن كذلك قم بتحويله إلى الصيغة المعيارية الثالثة عل شكل خطوات مع الرسم .

Course _No	Sec_ No	Dept	Credit _Hours	Course Level	Ins_id	Semester	Year	Date	Room _No	No_of _stu
رقم المقرر	الشعبة	القسم	الساعات المعتمدة	مستوى المقرر	رقم المدرس	الفصل الدراسي	السنة الدراسية	وقت الشعبة	القاعة الدراسية	عدد طلاب



المملكة العربية السعودية
المؤسسة العامة للتعليم الفني والتدريب المهني
الإدارة العامة لتصميم وتطوير المناهج

تصميم قواعد البيانات

تحويل نموذج الكيانات و العلاقات إلى نموذج علائقي

تحويل نموذج الكيانات و العلاقات إلى نموذج علائقي

٥

الجدارة:

القدرة على تحويل نموذج الكيانات و العلاقات إلى نموذج علائقي.

الأهداف:

أن يستطيع المتدرب تحويل نموذج الكيانات و العلاقات إلى نموذج علائقي.

مستوى الأداء المطلوب:

أن يتقن المتدرب عملية تحويل نموذج الكيانات والعلاقات إلى نموذج علائقي بنسبة ١٠٠٪.

الوقت المتوقع للتدريب:

ساعتان

الوسائل المساعدة:

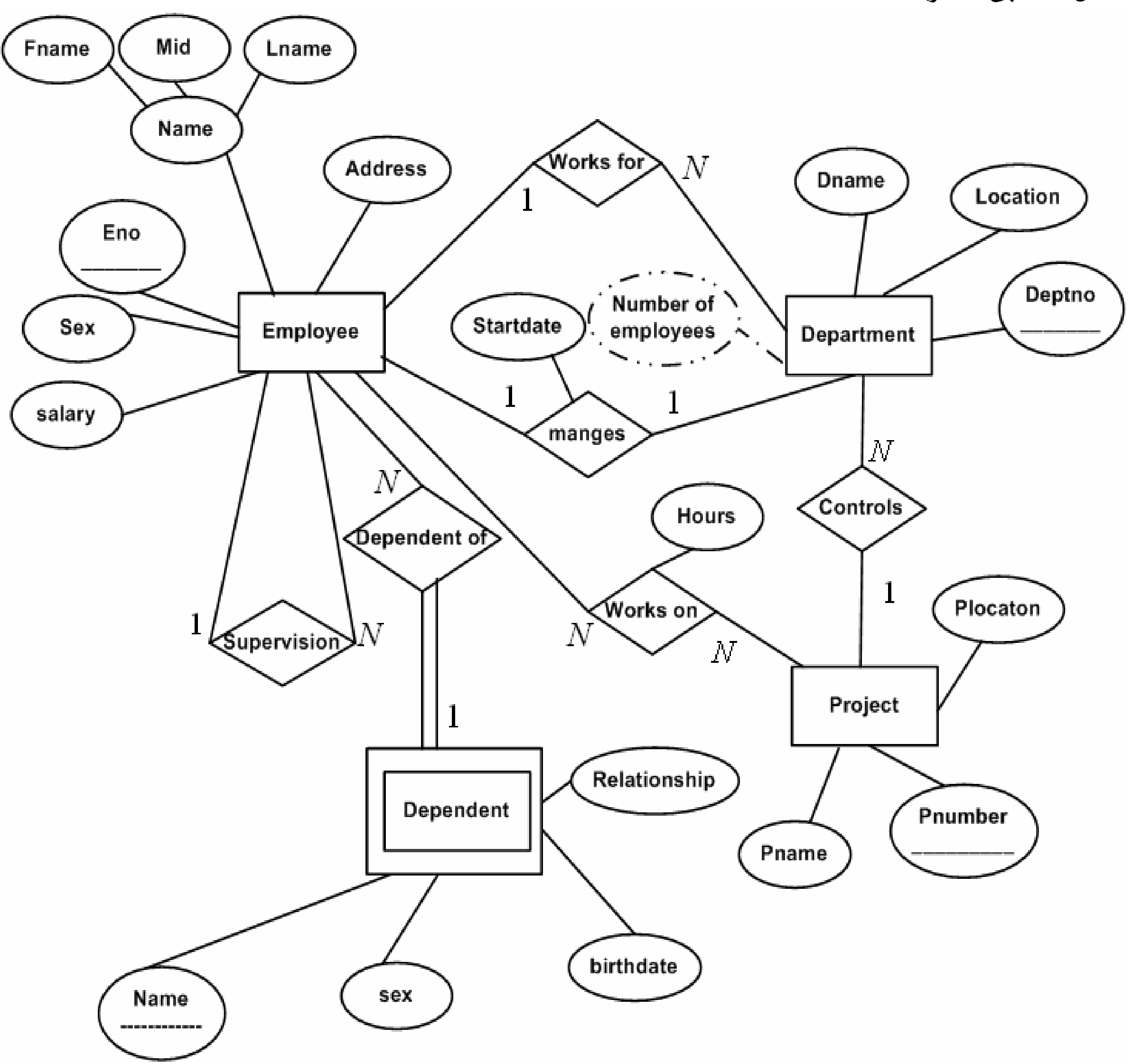
قلم + دفتر

متطلبات الجدارة:

أن يكون المتدرب قد أتقن الجدارة في الوحدات السابقة .

مقدمة

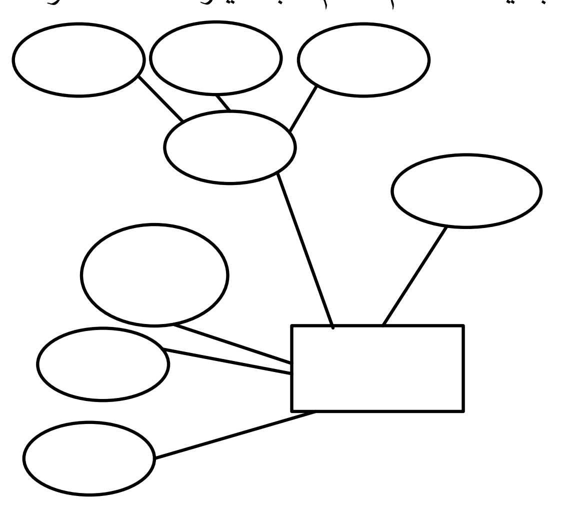
لتحويل عملية التصميم إلى قاعدة بيانات لابد في البداية من تحويل نموذج الكيانات والعلاقات) إلى نموذج علائقي حتى نسهل عملية تنفيذ هذا النموذج في قاعدة (إنشاء الجداول).وسنقوم في هذا الفصل بدراسة كيفية تحول نموذج المفاهيم(نموذج الكيانات والعلاقات) إلى نموذج علائقي مستخدمين المثال السابق للشركة .



والآن سنقوم بعدة خطوات لتحويل نموذج المفاهيم(نموذج الكيانات والعلاقات) إلى نموذج علائقي :

تحويل لكيانات :

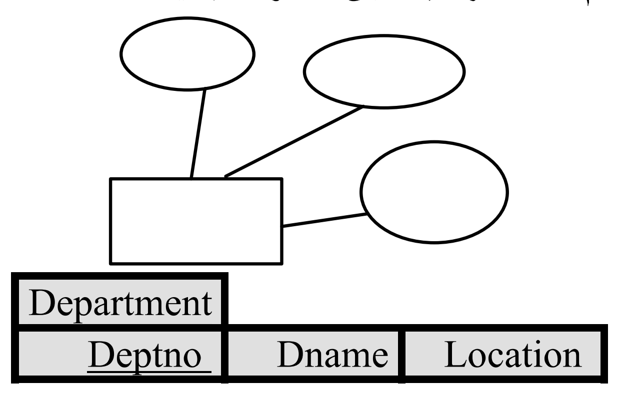
١ - لكل كيان(Entity (E في النموذج قم بإنشاء علاقة (Relation (R بحيث تحتوي العلاقة على جميع الصفات البسيطة غير المركبة وإذا كانت الصفات مركبة قم بتقسيمها إلى صفات بسيطة ،ثم قم باختيار صفة أو أكثر لتشكل المفتاح الرئيس للعلاقة .



نقوم بتحويلها لتصبح على الشكل التالي:

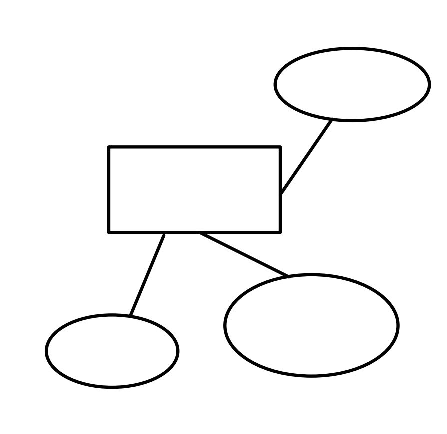
Employee						
<u>Eno</u> Fname	Mid	Lname	sex	Birthdate	Salary	

لاحظ أننا قمنا بتقسيم الاسم (صفة مركبة) إلى مكونات بسيطة .



Department		
<u>Deptno</u>	Dname	Location

لاحظ أننا لم نقم بإضافة عدد الموظفين (صفة مشتقة) ولكن يجب أن تأخذ بعين الاعتبار لإيجاد عدد الموظفين عن طريق بناء آلية استرجاع (Query).

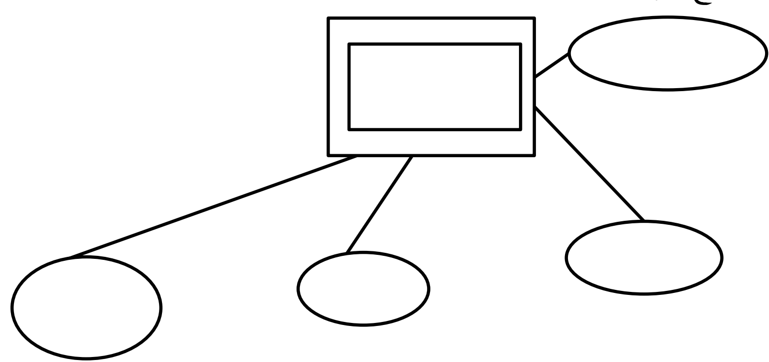




Project		
Pnumber	Pname	Plocation

تحويل لكيانات الضعيفة Weak Entity :

لكل كيان ضعيف (Weak Entity) في النموذج قم بإنشاء علاقة Relation (R) بحيث تحتوي العلاقة على جميع الصفات البسيطة غير المركبة وإذا كانت الصفات مركبة قم بتقسيمها إلى صفات بسيطة ،ثم قم باختيار إحدى الصفات مع المفتاح الرئيس للكيان الذي يتبع إليه الكيان الضعيف لتشكل المفتاح الرئيس للكيان،ثم قم بإنشاء مفتاح أجنبي ليشير إلى الكيان الذي يتبع الكيان الضعيف (المفتاح الرئيس لذلك الكيان).

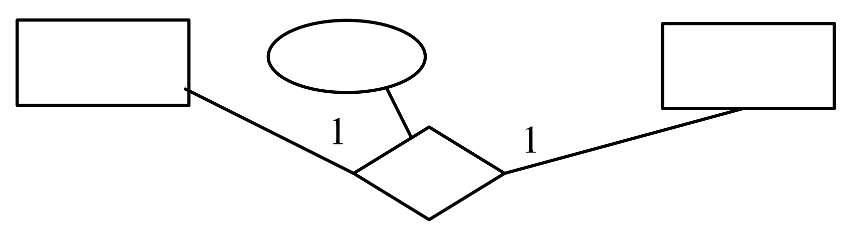


Dependent				
Eno	Name	Sex	Birthdate	Relationship

تحويل التشاركية: كما مر معنا سابقا فهناك ثلاثة أنواع من التشاركية علاقة واحد ـ واحد (١:١) وعلاقة واحد ـ متعدد (N:١) علاقة متعدد ـ متعدد (N:N) وسنقوم بعملية التحويل كل منها على النحو التالي :

١ - علاقة واحد ـ واحد (١:١)

لكل علاقة واحد ـ واحد (١:١) قم باختيار أحد الكيانين لتحتوي على مفتاح أجنبي ليشير إلى الكيان الآخر .

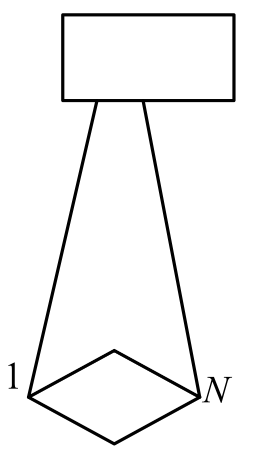


ففي هذه الحالة نقوم بإضافة صفة جديدة (Mgr) لتشير إلى الموظف الذي يتولى إدارة القسم (مفتاح أجنبي لجدول الموظفين) وكذلك إضافة تاريخ بداية إدارة هذا الموظف لذلك القسم .

Department				
Deptno	Dname	Location	Mgr	Startdate

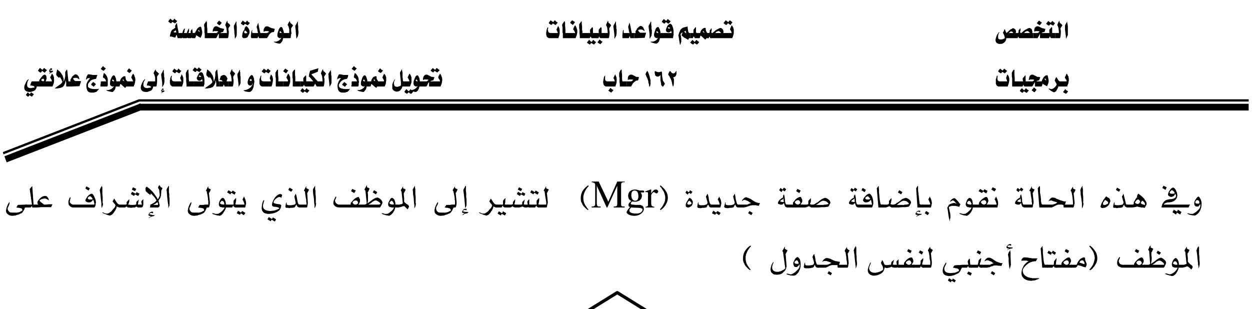
٢ - علاقة واحد ـ متعدد(N: 1)

لكل علاقة واحد ـ متعدد(N:1) قم بإضافة عمود(أعمدة) لتكون مفتاحا أجنبيا في جانب المتعدد(N) ليشير إلى المفتاح الرئيس في جانب الواحد(١) .

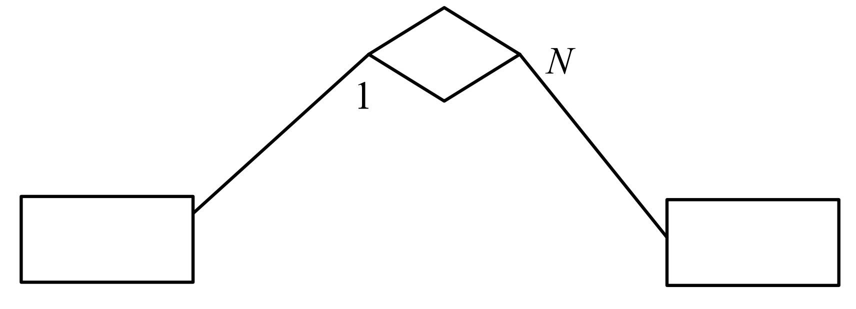


Employee							
Eno	Fname	Mid	Lname	sex	Birthdate	Salary	Mgr

Employee



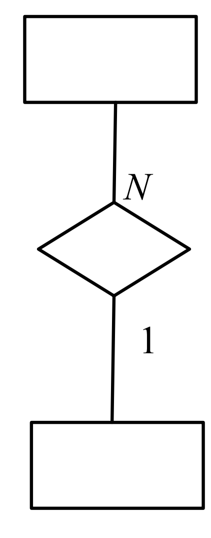
وفي هذه الحالة نقوم بإضافة صفة جديدة (Mgr) لتشير إلى الموظف الذي يتولى الإشراف على الموظف (مفتاح أجنبي لنفس الجدول)



Employee

Eno	Fname	Mid	Lname	sex	Birthdate	Salary	Mgr	Deptno

وفي هذه الحالة نقوم بإضافة صفة جديدة (Deptno) لتشير إلى القسم الذي يتبع إليه الموظف (مفتاح أجنبي لجدول الأقسام)



Project

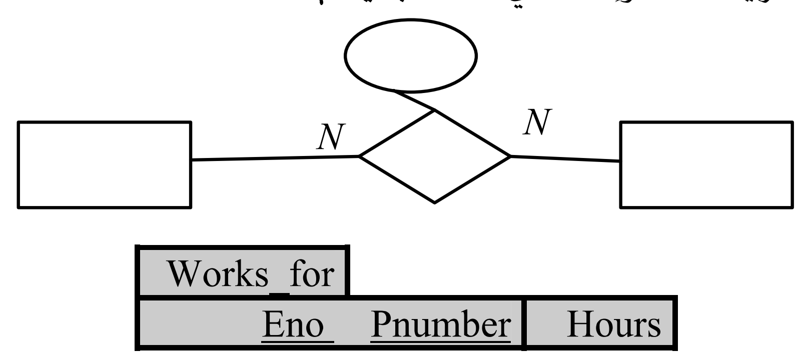
Pnumber	Pname	Plocation	Deptno

وفي هذه الحالة نقوم بإضافة صفة جديدة (Deptno) لتشير إلى القسم الذي يدير هذا المشروع (مفتاح أجنبي لجدول الأقسام).

Employee

٣ -علاقة متعدد ـ متعدد (N:N)

لكل علاقة متعدد ـ متعدد (N:N) قم بإنشاء علاقة جديد يكون المفتاح الرئيس لها عبارة عن دمج المفاتيح الرئيسة في طرفي العلاقة.وإضافة أي صفات جديد لهذه العلاقة



Works_for		
Eno	Pnumber	Hours

ففي هذه الحالة نقوم بإنشاء جدول جديد يحتوي (رمز المشروع،رقم الموظف ، عدد ساعات العمل) بحيث يشكل (رمز المشروع،رقم الموظف) المفتاح الرئيس للجدول وبنفس الوقت يكون رمز المشروع مفتاحاً أجنبياً لجدول المشاريع ، و رقم الموظف مفتاحاً أجنبياً لجدول الموظفين .

تحويل العلاقة بين الأنواع الفرعية(Subtype) **والأنواع العليا**(Super Type) ISA

وذلك عن طريق وضع المفتاح الرئيس في النوع الفرعي ليكون مفتاحا رئيسيا لهذا الجدول وفي نفس الوقت يكون مفتاحا أجنبيا للنوع الأعلى:

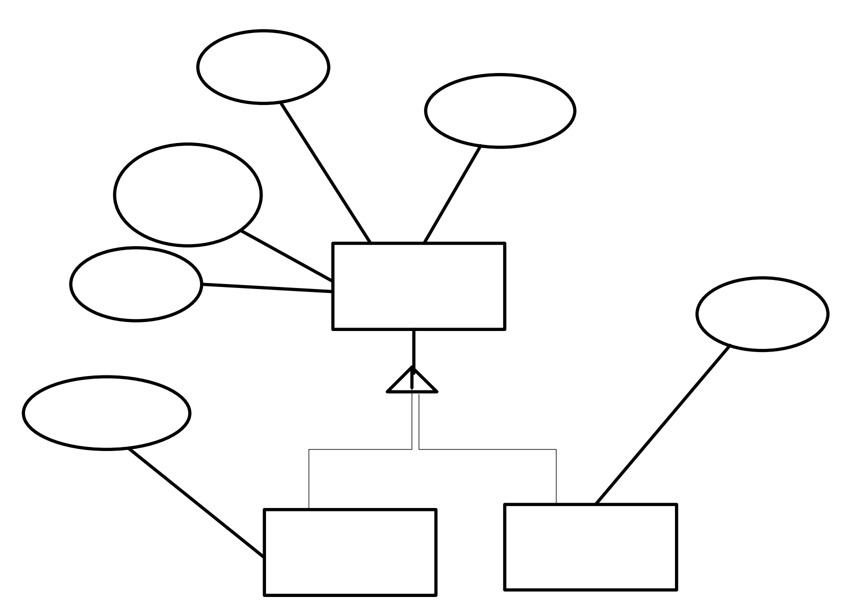
لنفرض أن لدينا نوعين من الموظفين

١ - موظف دائم يكون له راتب ثابت

٢ - موظف يعمل بالساعة ونسجل له أجرة العمل عن كل ساعة

فبالتالي يكون النموذج على الشكل التالي .

Employee



فناتج عملية التحويل يكون على النحو التالي:

S_Employee	
Eno	Salary

H_Employee	
Eno	Hour_Rate_

Eno

Sex

Hour_rate

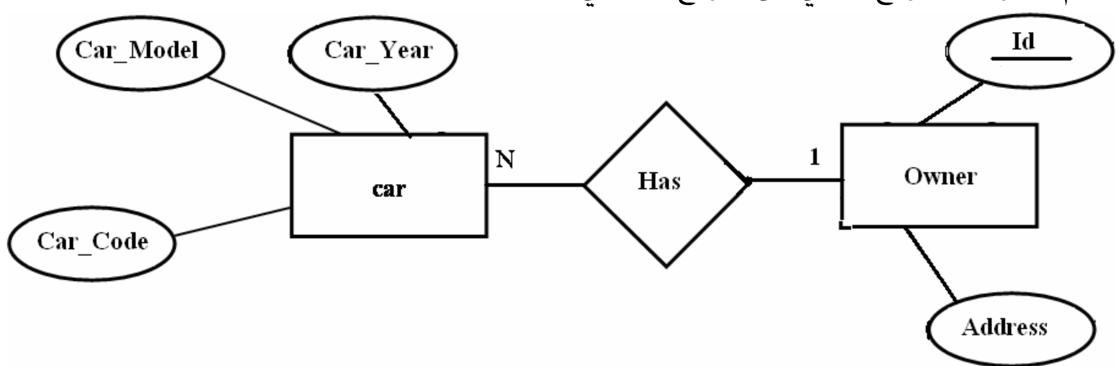
تمارين

١. ما الفائدة من تحويل نموذج الكيانات و العلاقات إلى نموذج علائقي؟

٢. بين باستخدام الرسم كيفية تحويل تشاركية N:N .

٣. بين باستخدام الرسم كيفية تحويل العلاقة بين الأنواع الفرعية(Subtype) والأنواع العليا(Super Type) ISA.

٤. قم بتحويل النموذج التالي إلى نموذج علائقي



٥. قم بتحويل النموذج الناتج من حل السؤال ٦ في الوحدة الثالثة إلى نموذج علائقي.

SQL (Structured Query Language)
لغة الاستفسار المهيكلة

- هي لغة تستخدم للاستفسار عن الجداول ضمن قاعدة البيانات بطريقة سريعة وسهلة.
- SQL تسمح لمستخدمين للوصول لبيانات في أنظمة إدارة قواعد البيانات العلائقية، مثل: Oracle , Sybase, Informix, Microsoft SQL Server, Access وغيرها.
- SQL لا تحتوي على أوامر تحكم مثل If .. then .. else أو Do … While . أي أنها ليست لغة إجرائية .
- SQL يمكن أن تستعمل بطريقتين :
  1- بطريقة تفاعلية عن طريق إدخال الجمل في الشاشة.
  2- عن طريق تضمين جمل SQL في لغة من لغات البرمجة (دلفي مثلاً).

لغة SQL تتضمن :

1- لغة تعريف البيانات DDL (Data Definition Language)
2- لغة معالجة البيانات DML (Data Manipulation Language) .

لغة تعريف البيانات DDL تسمح للمستخدم لإجراء المهام التالية :

- تكوين كائن قاعدة بيانات Create a database object.
- حذف كائن من قاعدة بيانات Drop a database object.
- تعديل كائن قاعدة بيانات Alter a database object.
- منح صلاحيات على كائن قاعدة بيانات Grant/Revoke privileges on database object.

لغة معالجة البيانات DML تسمح للمستخدم لإجراء المهام التالية :

- إضافة سجلات بيانات إلى الجدول **Insert** – to add rows of data to a table.
- حذف سجلات من الجدول **Delete** – to delete rows of data from a table.
- تعديل بيانات الجدول **Update** – to change data in a table.
- استرجاع سجلات بيانات من الجدول **Select** – to retrieve records from table/view.
- تخزين العمل بشكل نهائي على القرص **Commit work** – to make changes permanent.
- التراجع عن التغيرات منذ أخر تثبيت نهائي **Rollback** – to undo all changes since the last commit.

جملة الاستفسار Select :

الشكل العام لها:

Select [distinct] { * | [column [as new name]] [,...] } from table
[where <condition>]
[order by <column(s) [asc | desc] >]

الشكل المبسط لها :

Select <قائمة الأعمدة> from <اسم الجدول> ;

ملاحظة : كل جملة يجب أن تنتهي بفاصلة منقوطة.

لتنفيذ أوامر SQL سوف نستخدم قاعدة البيانات Oracle ، وبالتحديد سوف نستخدم SQL Plus 8.0 .

ولدخول إلى بيئة SQL Plus 8.0 نستخدم إحدى كلمات المرور التالية:

User name	password
Scott	tiger
System	manager

فمثلاً :

Log On

User Name: Scott

Password: *****

Host String:

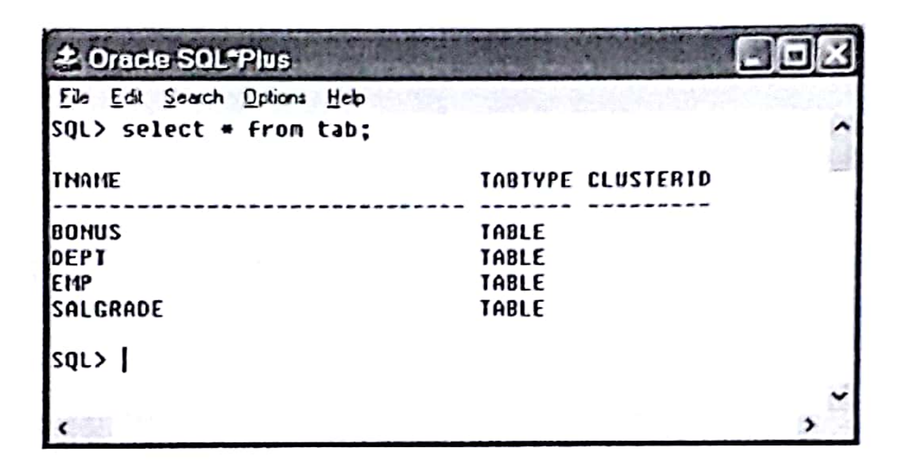
OK   Cancel

فعند استخدامنا لـSoctt/tiger نجد أن هناك عدة جداول مسبقة التعريف منها جدول الموظفين Emp ، وجدول الأقسام Dept وغيرها ، وسوف نستخدمها لإجراء الاستفسارات.

لاستعراض الجداول الموجدة نكتب التعليمة التالية :

```
select * from tab;
```

أي استعرض كل الجداول (Tab) المسموح باستعراضها من قبل Scott ، فتكون النتيجة كالتالي:



```
Oracle SQL*Plus
File Edit Search Options Help
SQL> select * from tab;

TNAME                          TABTYPE CLUSTERID
------------------------------ ------- ---------
BONUS                          TABLE
DEPT                           TABLE
EMP                            TABLE
SALGRADE                       TABLE

SQL> |
```

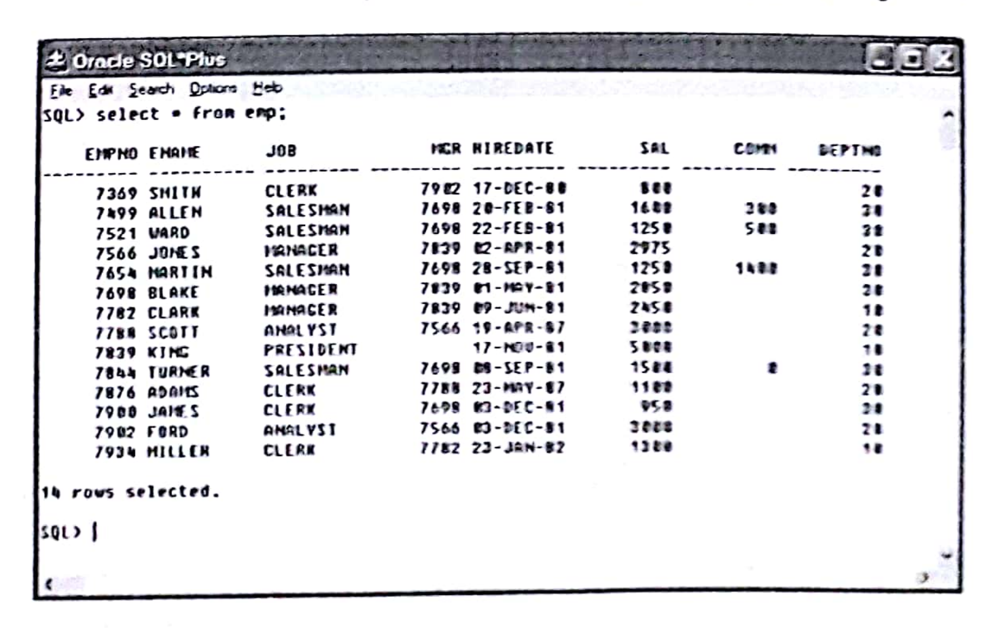
نجد أن هناك أربعة جداول يمكن التعامل معها.

مثال :

لاسترجاع كل الأعمدة من جدول الموظفين ***Emp*** نكتب التالي :

```
select * from emp;
```

حيث تعني النجمة (*) كل الحقول ، فتكون النتيجة كالتالي:



```
Oracle SQL*Plus
File Edit Search Options Help
SQL> select * from emp;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

14 rows selected.

SQL> |
```

ملاحظة :

1- لتنظيف شاشة SQL Plus 8.0 يمكن أن نكتب :

```
clear screen;
```

أو

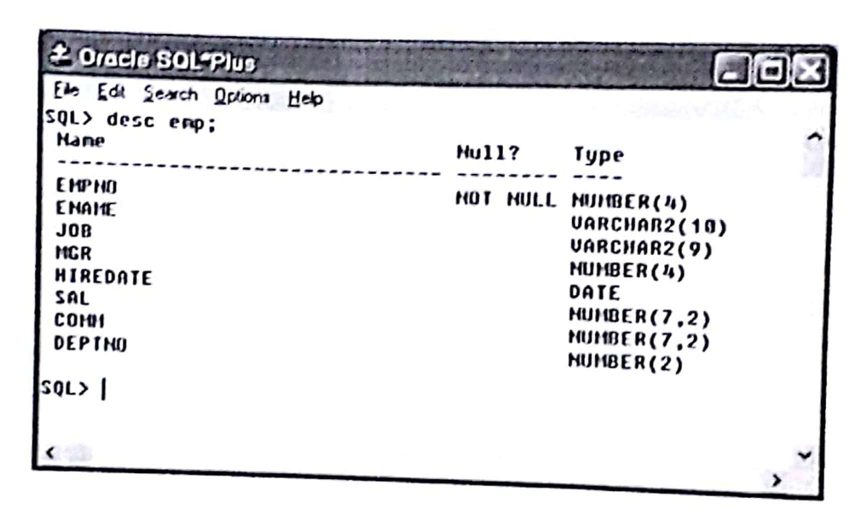
```
Cl scr;
```

2- لاستعراض هيكل أي جدول يمكن أن نستخدم الأتي :

```
Desc(ribe) Emp;
```

مثال :

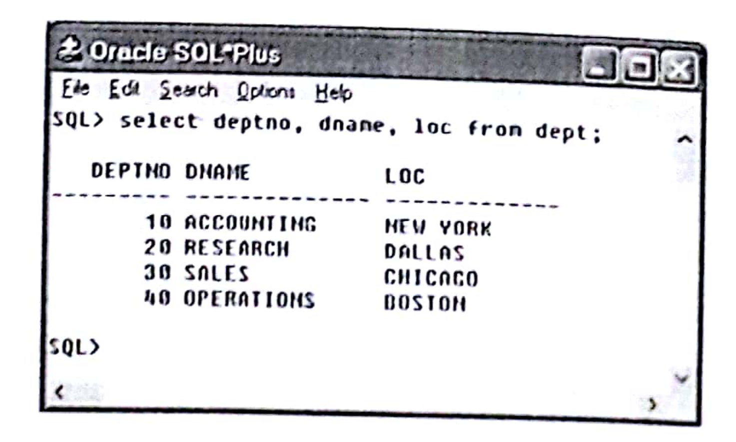
```
desc emp;
```



حيث يعني أن رقم الموظف (*EmpNo*) هو رقم ولا يمكن أن بلا قيمة (***not null***) وأن اسم الموظف (***Ename***) هو سلسلة تتكون من 10 رموز .... وهكذا. أما عن كيفية إنشاء جدول وتفاصيل تعريف الحقول به ستناولها لاحقاً.

يمكن أن نسرد أسماء الحقول كاملة بدل (*) فمثلا لإظهار كل حقول جدول الأقسام dept نكتب :

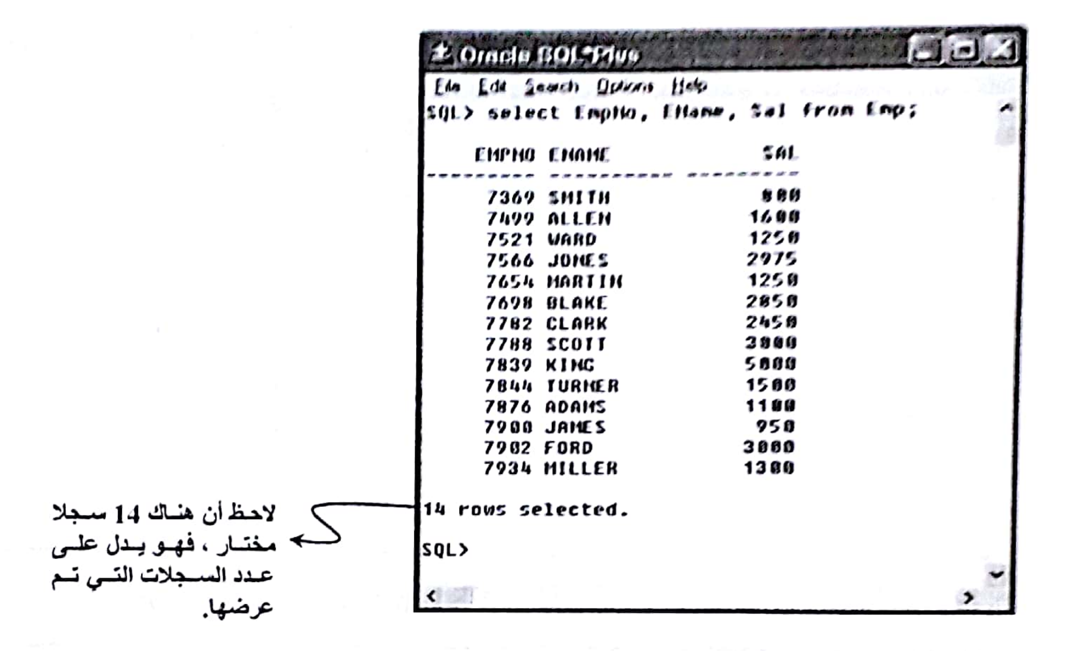
```
select deptno, dname, loc from dept;
```



مثال :

لاسترجاع حقول محددة فقط نكتب أسماء مفصولة بينها بفاصلة عادية ، فمثلا لاسترجاع أرقام (EmpNo) وأسماء (Ename) ومرتبات الموظفين السنوية (Sal) من جدول الموظفين (Emp) نكتب:

```
select EmpNo, EName, Sal from Emp;
```



```
Oracle SQL*Plus
File Edit Search Options Help
SQL> select EmpNo, EName, Sal from Emp;

    EMPNO ENAME             SAL
--------- ---------- ----------
     7369 SMITH             800
     7499 ALLEN            1600
     7521 WARD             1250
     7566 JONES            2975
     7654 MARTIN           1250
     7698 BLAKE            2850
     7782 CLARK            2450
     7788 SCOTT            3000
     7839 KING             5000
     7844 TURNER           1500
     7876 ADAMS            1100
     7900 JAMES             950
     7902 FORD             3000
     7934 MILLER           1300

14 rows selected.

SQL>
```


مثال :

لاسترجاع أرقام (EmpNo) وأسماء (Ename) وأرقام اقسام الموظفين (DeptNo) من جدول الموظفين (Emp) نكتب:

```
select EmpNo, EName, DeptNo from Emp;
```

```
Oracle SQL*Plus
File Edit Search Options Help
SQL> select EmpNo, EName, DeptNo from Emp;

    EMPNO ENAME          DEPTNO
--------- ---------- ----------
     7369 SMITH              20
     7499 ALLEN              30
     7521 WARD               30
     7566 JONES              20
     7654 MARTIN             30
     7698 BLAKE              30
     7782 CLARK              10
     7788 SCOTT              20
     7839 KING               10
     7844 TURNER             30
     7876 ADAMS              20
     7900 JAMES              30
     7902 FORD               20
     7934 MILLER             10

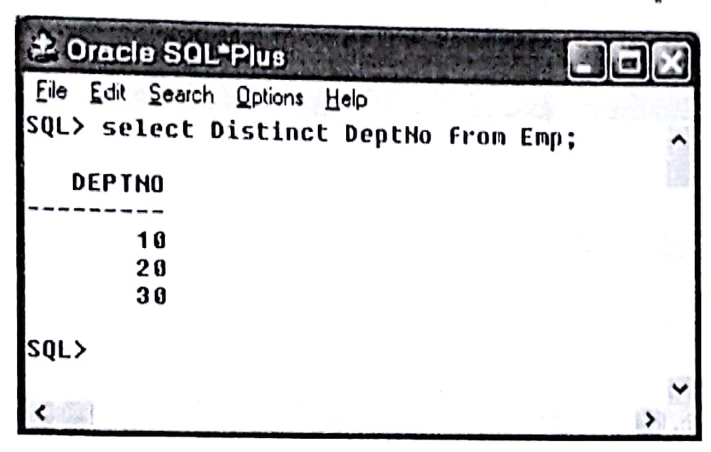
14 rows selected.

SQL> |
```

لاحظ أن أرقام الأقسام يوجد بها تكرار فمثلا لو أردنا أن نسترجع فقط ما هي الأقسام التي يعمل بها الموظفون يمكن أن نستخدم الكلمة **Distinct** والتي ترجع القيم بدون تكرار فمثلا إذا كتبنا الأتي :

```
select Distinct DeptNo from Emp;
```

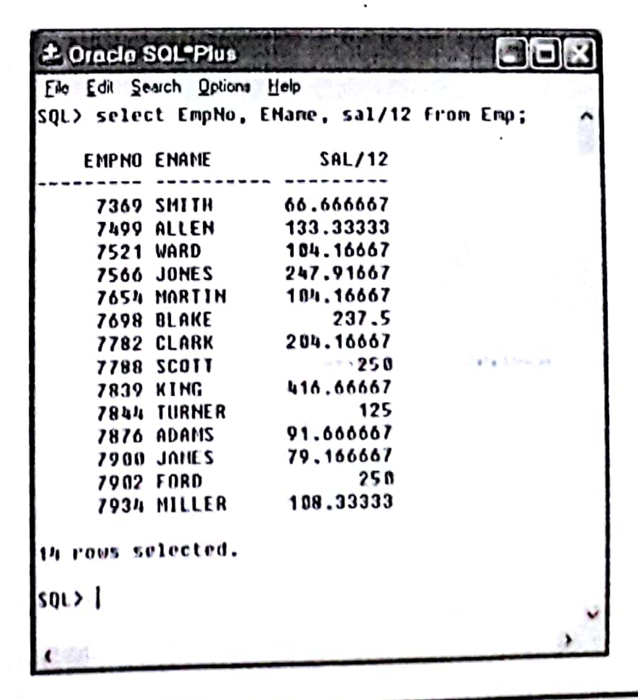
ستظهر فقط أرقام الأقسام التي يعمل بها الموظفون بدون تكرار (لاحظ أن الموظفون يعملون في الأقسام 10، 20 ، 30 فقط) كالتالي :



الحقول المحسوبة Calculated Fields :

لاحظ أن هناك حقل هو عمود المرتب السنوي Sal ، فلو أردنا أن نعرض قائمة بأرقام وأسماء الموظفين ومرتباتهم الشهري (أي المرتب السنوي مقسوما على 12) يمكن أن نكتب الأتي :

```
select EmpNo, EName, sal/12 from Emp;
```



حيث يمكن إجراء العمليات الحسابية الأربعة (+ ، - ، * ، /) على الأعمدة.

مثال :

تقرر إعطاء مكافأة لجميع الموظفين قدرها 300 دينار مضافة إلى مرتباتهم الأصلية ، فاعرض قائمة بأسماء ومرتبات الموظفين قبل و بعد المكافأة ، نكتب التالي :

```
Select EName, Sal , Sal+300 From Emp;
```

المرتب قبل المكافأة — المرتب بعد المكافأة

```
Oracle SQL*Plus
File Edit Search Options Help
SQL> Select EName, Sal , Sal+300 From Emp;

ENAME            SAL    SAL+300
---------- ---------- ----------
SMITH            800       1100
ALLEN           1600       1900
WARD            1250       1550
JONES           2975       3275
MARTIN          1250       1550
BLAKE           2850       3150
CLARK           2450       2750
SCOTT           3000       3300
KING            5000       5300
TURNER          1500       1800
ADAMS           1100       1400
JAMES            950       1250
FORD            3000       3300
MILLER          1300       1600

14 rows selected.

SQL> |
```

لاحظ أن رأس العمود للمرتب بعد المكافأة هي "Sal + 300" حيث بها بعض الغموض فلجعلها أكثر وضوحا من الممكن أن نجعلها "New Salary" باستخدام جملة AS كالتالي :

```
Select EName, Sal , Sal+300 As "New Salary" From Emp;
```

```
Oracle SQL*Plus
File Edit Search Options Help
SQL> Select EName, Sal , Sal+300 As "New Salary" From Emp;

ENAME            SAL New Salary
---------- ---------- ----------
SMITH            800       1100
ALLEN           1600       1900
WARD            1250       1550
JONES           2975       3275
MARTIN          1250       1550
BLAKE           2850       3150
CLARK           2450       2750
SCOTT           3000       3300
KING            5000       5300
TURNER          1500       1800
ADAMS           1100       1400
JAMES            950       1250
FORD            3000       3300
MILLER          1300       1600

14 rows selected.

SQL> |
```

إذا لا يوجد بالتسمية الجديدة فراغ يمكن كتابتها بدون علامتي التنصيص كالتالي :

```
Select EName, Sal , Sal+300 As New_Salary From Emp;
```

إلى حدا الآن نحن نقوم بتحديد أعمدة معينة من الجدول ، ولكن ما لو أردنا أن نحدد سجلات (صفوف) معينة بناء على شرط معين فإننا نستخدم جملة Where ونضع الشرط المطلوب بعدها، فمثلا لو أردنا استرجاع كل الحقول من جدول الموظفين العاملين في القسم رقم 20 نكتب الآتي :

```
Select * from Emp
  Where DeptNo = 20;
```

```
Oracle SQL*Plus
File Edit Search Options Help
SQL> Select * from Emp
  2  Where DeptNo = 20;

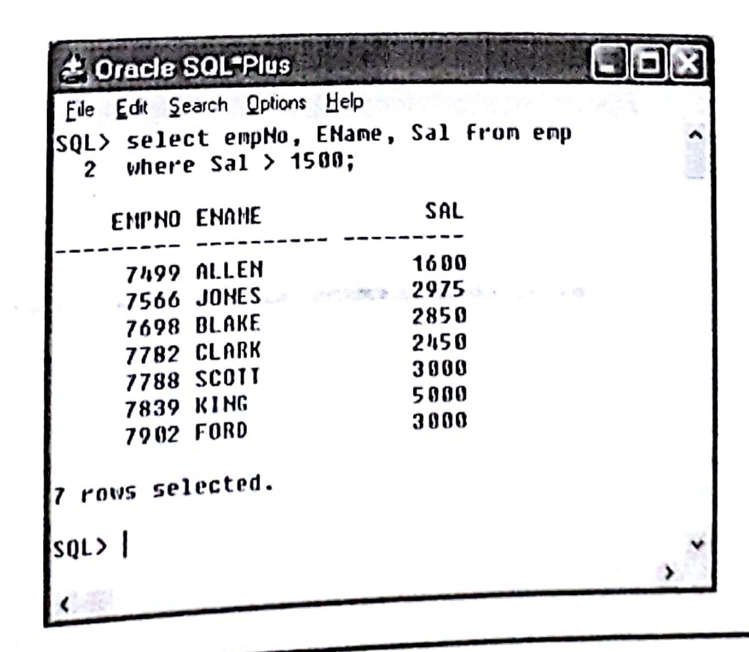
     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20

SQL>
```

مثال أخر :

استرجع أرقام وأسماء ومرتبات الموظفين والذين مرتباتهم تزيد عن 1500 ، نكتب الآتي :

```
select empNo, EName, Sal from emp
where Sal > 1500;
```



```
Oracle SQL*Plus
File Edit Search Options Help
SQL> select empNo, EName, Sal from emp
  2  where Sal > 1500;

     EMPNO ENAME             SAL
---------- ---------- ----------
      7499 ALLEN            1600
      7566 JONES            2975
      7698 BLAKE            2850
      7782 CLARK            2450
      7788 SCOTT            3000
      7839 KING             5000
      7902 FORD             3000

7 rows selected.

SQL> |
```

مؤثرات المقارنة التي يمكن أن تستخدم هي :

اصغر من	<	يساوي	=
اصغر من أو يساوي	<=	اكبر من	>
لا يساوي ^= أو <> أو (ISO Standard) !=		اكبر من أو يساوي	>=

كما يمكن أن نستخدم المعاملات (و - And) ، (أو - Or) ، (نفي - Not) لتكوين تعبير معقد.

مثال :

استرجع أسماء الموظفين ومرتباتهم مع استحداث عمود جديد يمثل الضريبة Tax والتي تمثل 5% من قيمة المرتب للموظفين التي مرتباتهم اكبر من 2000 ويعملون بالقسم رقم 20.

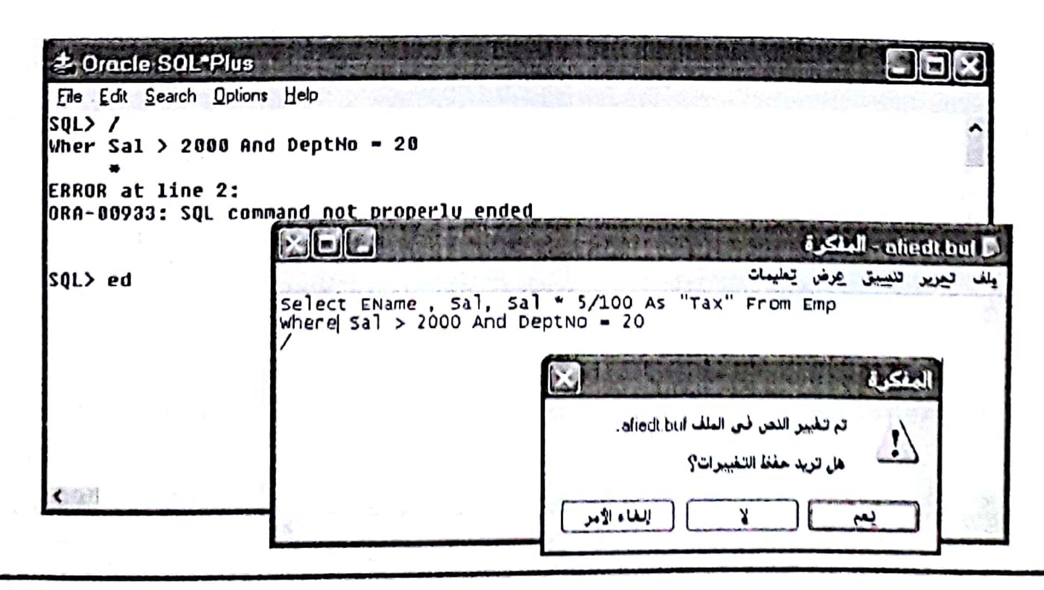
```
Select EName , Sal, Sal * 5/100 As "Tax" From Emp
Where Sal > 2000 And DeptNo = 20;
```

```
Oracle SQL*Plus
File Edit Search Options Help
SQL> Select EName , Sal, Sal * 5/100 As "Tax" From Emp
  2  Where Sal > 2000 And DeptNo = 20;

ENAME             SAL        Tax
---------- --------- ---------
JONES            2975     148.75
SCOTT            3000        150
FORD             3000        150

SQL>
```

لنفرض انك أخطأت في كتابة الجملة ، فبدلا من إعادة كتابتها من جديد يمكن الذهاب إلى المفكرة بكتابة الأمر Ed لتجد أخر تعليمة قمت بإدخالها موجودة هناك ، فقم بتعديل الخطأ ثم خزن التعديلات ومن محث SQL اكتب فقط '/' . ليتم تنفيذ التعديلات الجديدة.



مثال :

استرجع كل سجلات الموظفين الذين مرتباتهم تكون اما 3000 أو 5000.

```
Select * From Emp
Where Sal = 3000 OR Sal =5000;
```

```
Oracle SQL*Plus
File Edit Search Options Help
SQL> Select * From Emp
  2  Where Sal = 3000 OR Sal = 5000;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20

SQL>
```

مثال :

استرجع كل سجلات الموظفين الذين مرتباتهم بين 1500 و 3000.

```
Select * From Emp
Where Sal >= 1500 and Sal <=3000;
```

أو يمكن استخدام (Between… and) ، حيث أن 1500 و 3000 هي ضمن المجال كالتالي:

```
Select * From Emp
Where Sal between 1500 and 3000;
```

```
Oracle SQL*Plus
File Edit Search Options Help
SQL> Select * From Emp
  2  Where Sal between 1500 and 3000;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20

7 rows selected.

SQL> |
```

مثال :

لو أردنا العكس ، أي استرجع كل سجلات الموظفين الذين مرتباتهم ليست بين 1500 و 3000.

هنا يمكن أن نستخدم المعامل Not كالتالي :

```
Select * From Emp
Where Sal Not between 1500 and 3000;
```

```
Oracle SQL*Plus
File  Edit  Search  Options  Help
SQL> Select * From Emp
  2  Where Sal Not between 1500 and 3000;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

7 rows selected.

SQL> |
```

مثال :

استرجع كل سجلات الموظفين الذين لهم وظيفة مدير 'MANAGER' أو مدير مبيعات 'SALESMAN'.

هنا يمكن أن نستخدم التالي :

```
Select * From Emp
Where Job = 'MANAGER' OR Job = 'SALESMAN';
```

أو يمكن استخدام اختبار العضوية (IN) لاختبار إذا كانت القيمة مساوية لأي من قائمة القيم المسرودة ، وهنا هي 'MANAGER' أو 'SALESMAN' .

```
Select * From Emp
Where Job in ('MANAGER','SALESMAN');
```

```
Oracle SQL*Plus
File  Edit  Search  Options  Help
SQL> Select * From Emp
  2  Where Job in ('MANAGER','SALESMAN');

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30

7 rows selected.

SQL> |
```

مثال :

وبــالعكس يمكــن اســترجاع كــل ســجلات المــوظفين الــذين لهــم ليســوا مــدراء 'MANAGER' أو مدراء مبيعات 'SALESMAN'.

هنا يمكن أن نستخدم **Not** كالتالي :

```
Select * From Emp
Where Job Not in ('MANAGER','SALESMAN');
```

```
Oracle SQL*Plus
File Edit Search Options Help
SQL> Select * From Emp
  2  Where Job Not in ('MANAGER','SALESMAN');

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

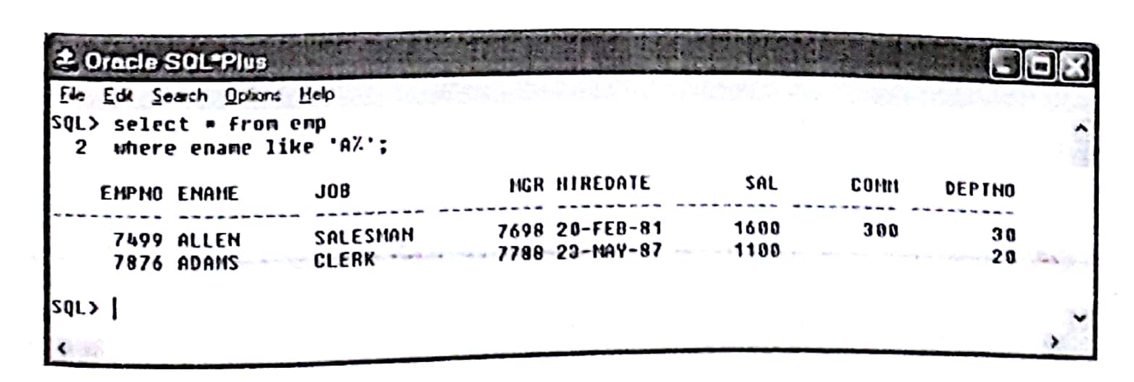
7 rows selected.

SQL> |
```

مثال :

لاسترجع كل الموظفين الذين يبدأ أسمائهم بـ'A' يمكن أن نكتب الآتي :

```
select * from emp
where ename like 'A%';
```



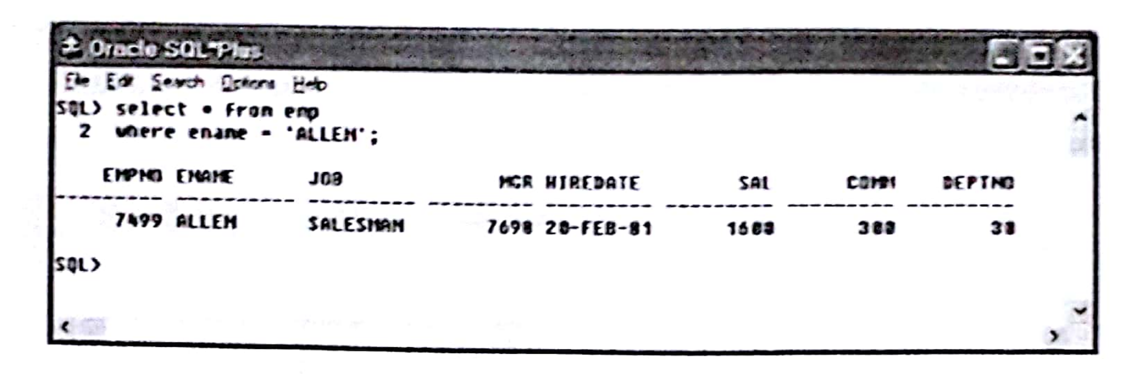
```
Oracle SQL*Plus
File Edit Search Options Help
SQL> select * from emp
  2  where ename like 'A%';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20

SQL> |
```

لاحظ هنا استخدمنا ***Like*** ولم نستخدم "=" ، فإذا استخدمنا "=" فان النتيجة تعني الكلمة تماما فمثلا إذا أردنا أن نبحث عن موظف اسمه 'ALLEN' فإننا نكتب الآتي :

```
select * from emp
where ename = 'ALLEN';
```


Oracle SQL*Plus

File Edit Search Options Help

```
SQL> select * from emp
  2  where ename = 'ALLEN';
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30

SQL>

فهنا استرجع السجل المطابق للكلمة تماما ، كما أنه حساس لحالة الأحرف أي إذا كتبنا الحروف الصغيرة بدلاً من الكبيرة فان النتيجة سوف لن يرجع أي سجل.

عرفنا معني Like ولكن ما معني "%" ؟

"%" تعني أي تسلسل من الرموز (حروف ، ارقام ، علامات) ، كما يوجد لنا الرمز (_) الشرطة التحتية والذي يمثل رمز واحد فقط .

فمثلا :

'A%A' : يعني أن تبدأ السلسلة الرمزية بـ'A' ثم أي تسلسل من الرموز ثم تنتهي بـ'A'.

'A_Z' : يعني أن تبدأ السلسلة الرمزية بـ'A' ثم أي رمز أخر (واحد فقط) ثم تنتهي بـ'Z'.

'A_B%M' : يعني أن تبدأ السلسلة الرمزية بـ'A' ثم أي رمز أخر (واحد فقط) ثم أي تسلسل من الرموز ثم تنتهي بـ'M'.

'%A%' : يعني أن تحتوي السلسلة الرمزية ضمنها الحرف 'A' .

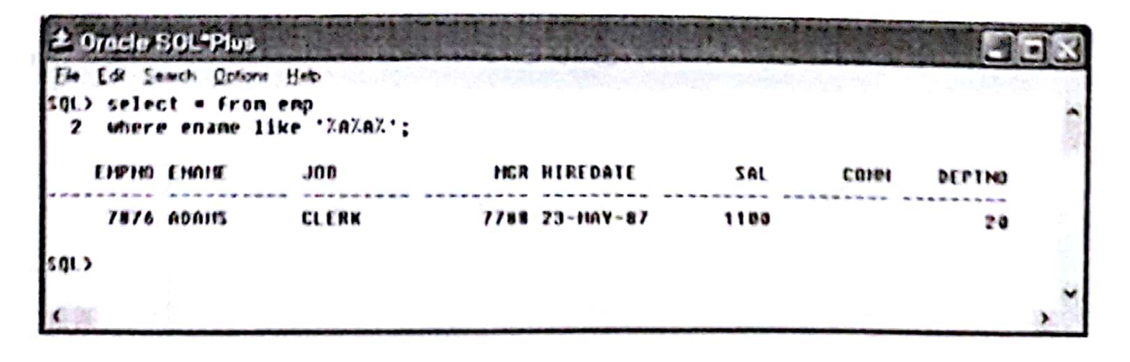
'%A' : يعني أن تنتهي السلسلة الرمزية بالحرف 'A' .

مثال :

استرجع الموظفين الذين تحتوي أسمائهم على حرفين من 'A' .

نكتب الآتي :

```
select * from emp
where ename like '%A%A%'
```


Oracle SQL*Plus

File Edit Search Options Help

```
SQL> select * from emp
  2  where ename like '%A%A%';
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20

SQL>

مثال :

استرجع أسماء الموظفين ومرتباتهم الذين لا تحتوي أسمائهم على حرف 'A' .

نكتب الآتي :

```
select * from emp
where ename NOT like '%A%'
```

```
Oracle SQL*Plus
File Edit Search Options Help
SQL> select * from emp
  2  where ename NOT like '%A%';

    EMPNO ENAME      JOB              MGR HIREDATE        SAL       COMM     DEPTNO
--------- ---------- --------- --------- --------- --------- --------- ---------
     7369 SMITH      CLERK           7902 17-DEC-80       800                  20
     7566 JONES      MANAGER         7839 02-APR-81      2975                  20
     7788 SCOTT      ANALYST         7566 19-APR-87      3000                  20
     7839 KING       PRESIDENT            17-NOV-81      5000                  10
     7844 TURNER     SALESMAN        7698 08-SEP-81      1500          0       30
     7902 FORD       ANALYST         7566 03-DEC-81      3000                  20
     7934 MILLER     CLERK           7782 23-JAN-82      1300                  10

7 rows selected.

SQL> |
```

لاحظ استخدام (*NOT Like*) والتي تعني "لا يشبه" .

القيمة Null : تعني أن القيمة غير معروفة أو معينة أو غير متاحة ، لذلك فان (=) لا يمكن أن تستخدم هنا لأنها لا تساوي أي قيمة ويمكن استخدمها بالصيغة :

Is Not Null   أو   Is null

مثال :

لاسترجاع كل من حقول أسم الموظف EName والمرتب Sal والعمولة Comm فقط للذين تم تعيين قيمة لحقل COMM ، فنكتب الآتي :

```
select Ename, Sal , Comm From emp
where comm is NOT null;
```

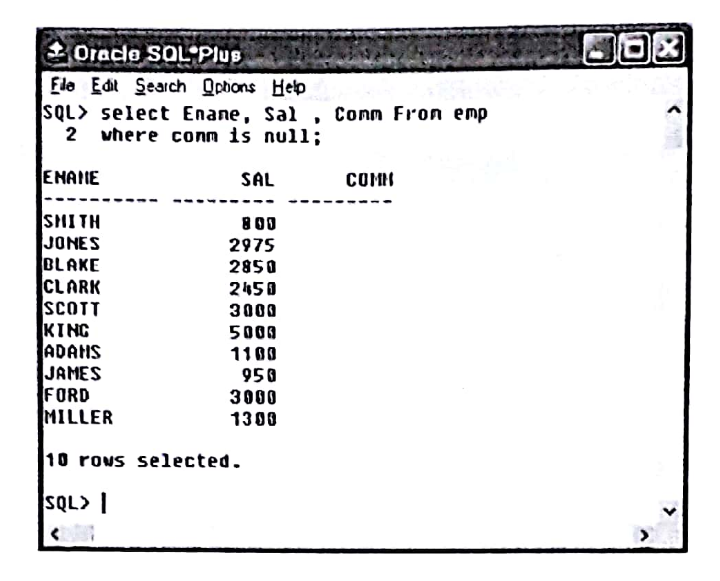
```
Oracle SQL*Plus
File Edit Search Options Help
SQL> select Ename, Sal , Comm From emp
  2  where comm is NOT null;

ENAME            SAL       COMM
---------- --------- ---------
ALLEN           1600       300
WARD            1250       500
MARTIN          1250      1400
TURNER          1500         0

SQL>
```

أما لاسترجاع كل الموظفين والذين لم يتم تعيين قيمة لهم ضمن حقل الـ Comm نكتب الآتي :

```
select Ename, Sal , Comm From emp
where comm is null;
```



```
SQL> select Ename, Sal , Comm From emp
  2  where comm is null;

ENAME             SAL       COMM
---------- --------- ---------
SMITH             800
JONES            2975
BLAKE            2850
CLARK            2450
SCOTT            3000
KING             5000
ADAMS            1100
JAMES             950
FORD             3000
MILLER           1300

10 rows selected.

SQL>
```

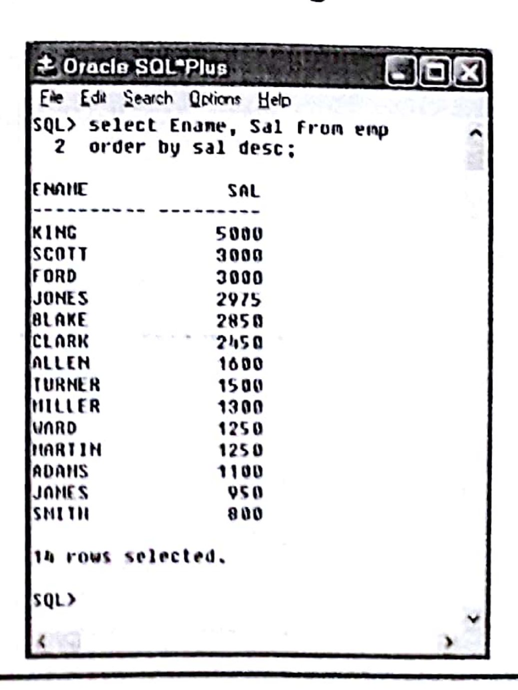
الترتيب Sorting :

يمكن أن نظهر النتائج مرتبة ضمن ترتيب معين باستخدام جملة Order by والتي تسمح لنا بترتيب الصفوف إما تصاعديا ASC (*Ascending*) أو تنازليا DESC (*Descending*) اعتمادا على عمودا واحدة أو أي تشكيلية من الأعمدة بغض النظر عن أي عمود قد يظهر ضمن النتيجة وعادة ما تظهر في آخر جملة Select .

مثال :

لاسترجاع أسماء الموظفين ومرتباتهم مرتبة تنازليا حسب المرتب نكتب الآتي :

```
select Ename, Sal from emp
order by sal desc;
```



```
SQL> select Ename, Sal from emp
  2  order by sal desc;

ENAME             SAL
---------- ---------
KING             5000
SCOTT            3000
FORD             3000
JONES            2975
BLAKE            2850
CLARK            2450
ALLEN            1600
TURNER           1500
MILLER           1300
WARD             1250
MARTIN           1250
ADAMS            1100
JAMES             950
SMITH             800

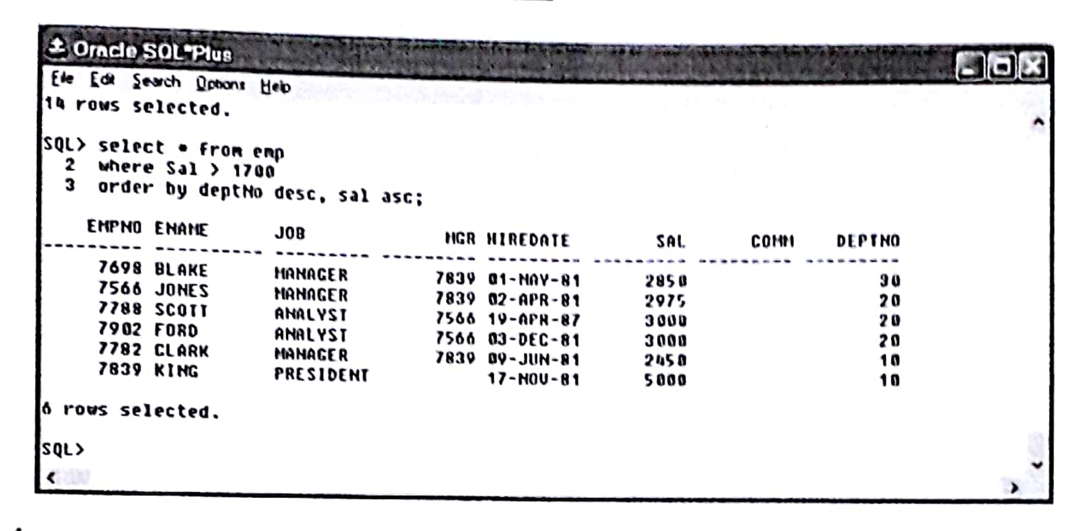
14 rows selected.

SQL>
```

مثال :

لاسترجاع الموظفين والذين مرتباتهم تزيد عن 1700 مرتبين تنازليا على أساس رقم القسم وتصاعديا على أساس الراتب نكتب الآتي :

```
select * from emp
where Sal > 1700
order by deptNo desc, sal asc;
```



```
Oracle SQL*Plus
File Edit Search Options Help
14 rows selected.

SQL> select * from emp
  2  where Sal > 1700
  3  order by deptNo desc, sal asc;

    EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
--------- ---------- --------- --------- --------- --------- --------- ---------
     7698 BLAKE      MANAGER         7839 01-MAY-81      2850                   30
     7566 JONES      MANAGER         7839 02-APR-81      2975                   20
     7788 SCOTT      ANALYST         7566 19-APR-87      3000                   20
     7902 FORD       ANALYST         7566 03-DEC-81      3000                   20
     7782 CLARK      MANAGER         7839 09-JUN-81      2450                   10
     7839 KING       PRESIDENT            17-NOV-81      5000                   10

6 rows selected.

SQL>
```

ملاحظة :

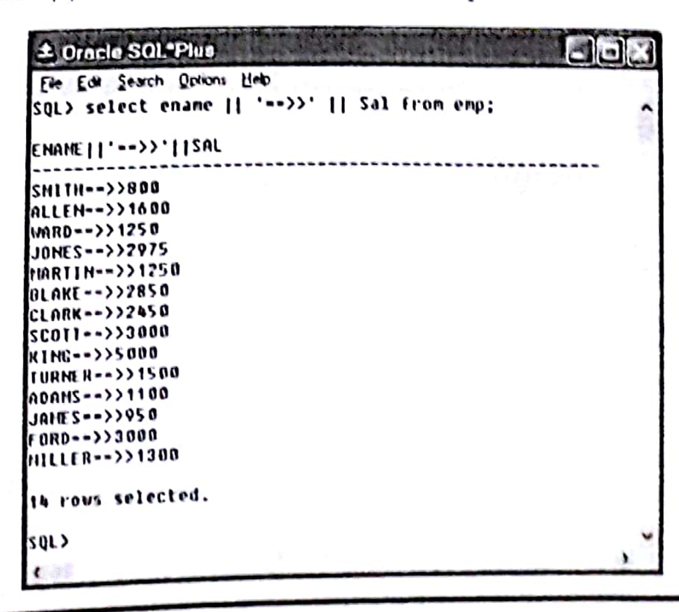
لعرض حقلين أو أكثر مندمجين في النتيجة كأنها حقل واحد نستخدم " || " فمثلا لإظهار اسم الموظف ومرتبه كحقل واحد نكتب:

```
select ename || Sal from emp;
```

مثال :

لاسترجاع الموظفين ومرتباتهم بحيث تظهر النتيجة على سبيل المثال كالتالي : `Ali==>>2000`

```
select ename || '==>>' || Sal from emp;
```



```
Oracle SQL*Plus
File Edit Search Options Help
SQL> select ename || '==>>' || Sal from emp;

ENAME||'==>>'||SAL
------------------------------------------------------------
SMITH==>>800
ALLEN==>>1600
WARD==>>1250
JONES==>>2975
MARTIN==>>1250
BLAKE==>>2850
CLARK==>>2450
SCOTT==>>3000
KING==>>5000
TURNER==>>1500
ADAMS==>>1100
JAMES==>>950
FORD==>>3000
MILLER==>>1300

14 rows selected.

SQL>
```

دوال التجميع Aggregate Functions:

هي دوال إحصائية يمكن أن تطبق على عمود واحد وترجع قيمة مفردة. وهي :

عدد القيم	Count
المجموع	Sum
المتوسط الحسابي	Avg
أصغر قيمة	Min
أكبر قيمة	Max

Count ، Min ، Max : يمكن أن تطبق على كل من الحقول الرقمية أو غير الرقمية .

Sum و Avg : تستعمل مع الحقول الرقمية فقط .

باستثناء Count(*) كل الدوال تحذف كل السطور التي تحتوي على Null من الحقل أولا ثم تطبق على كل القيم عدا Null . تذكر أن الدوال التجميعية لا يمكن أن تستخدم ضمن جملة where .

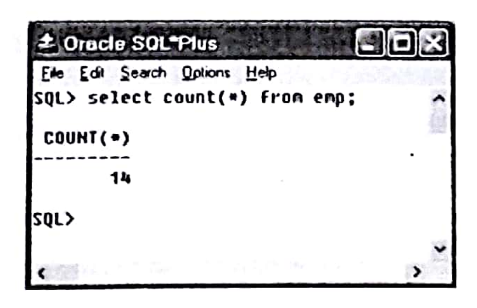
دالة العدد Count :

تقوم بحساب عدد القيم بما فيها null .

مثال :

استرجع عدد الموظفين ضمن جدول emp .

```
select count(*) from emp;
```



```
SQL> select count(*) from emp;

 COUNT(*)
---------
       14

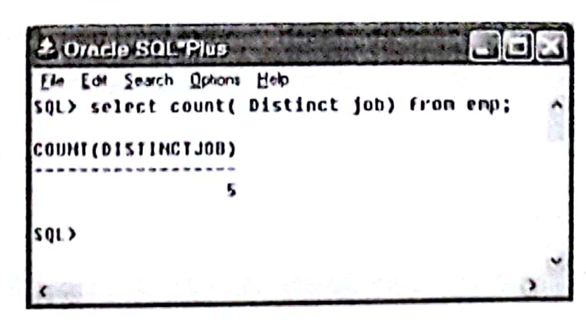
SQL>
```

مثال :

كم عنوان وظيفة مختلف مخزنة ضمن جدول emp .

نكتب الآتي :

```
select count( Distinct job) from emp;
```



```
SQL> select count( Distinct job) from emp;

COUNT(DISTINCTJOB)
------------------
                 5

SQL>
```

لاحظ انه عندما استخدمنا `Count(*)` قام بحساب عدد السجلات وهي 14 سجل.
لاحظ أنه عندما استخدمنا كلمة `Distinct` قام باسترجاع الوظائف المختلفة أولا بدون تكرار ثم قام بعد هذه السجلات الفريدة وهي خمسة أنواع من الوظائف.

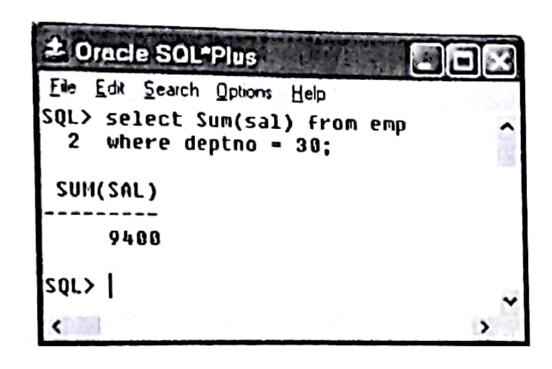
دالة المجموع Sum :

تقوم بحساب مجموع القيم وتطبق فقط على نوع البيانات الرقمي.

مثال :

لإيجاد مجموع مرتبات الموظفين العاملين بالقسم رقم 30 نكتب الآتي :

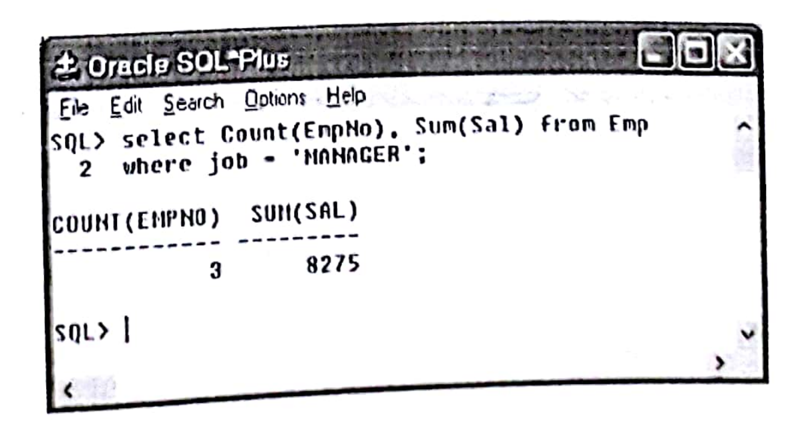
```
select Sum(sal) from emp
where deptno = 30;
```



مثال :

لإيجاد عدد المدراء ومجموع مرتباتهم نكتب الآتي :

```
select Count(EmpNo), Sum(Sal) from Emp
where job = 'MANAGER';
```



دالة أصغر وأكبر قيمة Min & Max :

تقوم بإيجاد اصغر واكبر قيمة على التوالي.

مثال :

لإيجاد اصغر مرتب نكتب الآتي :

```
Select Min(sal) from emp;
```

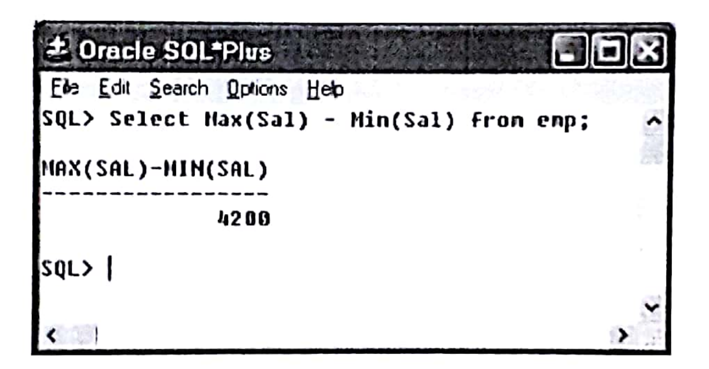
أما لإيجاد أكبر مرتب فنكتب الآتي :

```
Select Max(sal) from emp;
```

مثال :

لإيجاد الفرق بين أكبر واصغر مرتب نكتب الآتي :

```
Select Max(Sal) - Min(Sal) from emp
```


```
SQL> Select Max(Sal) - Min(Sal) from emp;

MAX(SAL)-MIN(SAL)
-----------------
             4200

SQL>
```

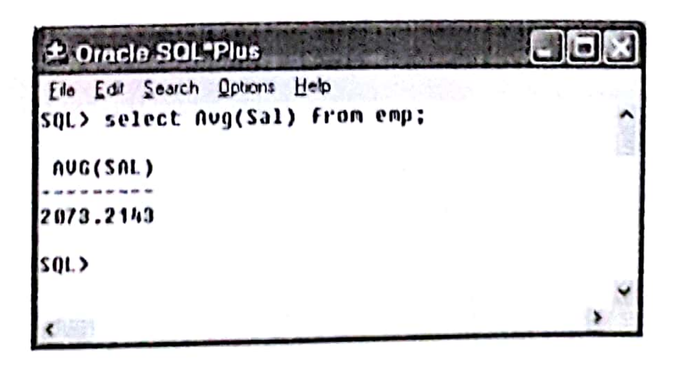
دالة المعدل Avg:

تقوم بإيجاد المتوسط الحسابي لقيم عمود ما ، وهي تطبق على الحقول الرقمية فقط.

مثال :

لإيجاد المتوسط الحسابي لمرتبات الموظفين ، نكتب الآتي :

```
select Avg(Sal) from emp;
```


```
SQL> select Avg(Sal) from emp;

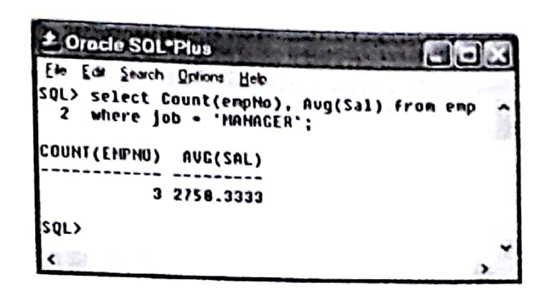
 AVG(SAL)
---------
2073.2143

SQL>
```

مثال :

لإيجاد عدد و المتوسط الحسابي لمرتبات المدراء ، نكتب الآتي :

```
select Count(empNo), Avg(Sal) from emp
where job = 'MANAGER';
```



```
Oracle SQL*Plus
File Edit Search Options Help
SQL> select Count(empNo), Avg(Sal) from emp
  2  where job = 'MANAGER';

COUNT(EMPNO)  AVG(SAL)
------------ ---------
           3 2758.3333

SQL>
```

لاحظ أننا نستخدم الحروف الكبيرة في كتابة كلمة 'MANAGER' مثلا لكي تكون النتيجة صحيحة لأنها مخزنة على أساس حروف كبيرة ، ولكن أحيانا لا نعرف الكلمات المخزنة هل هي بحروف كبيرة أو صغيرة ، لذلك يمكن استخدام إحدى الدالتين (( )Lower أو ( )Upper) لتحويل القيمة من العمود المقارن لحروف صغيرة (Lower) أو حروف كبيرة (Upper) كالتالي:

```
Select * from Emp
where Lower(job) = 'manager';
```

حيث تقوم الدالة Lower(job) بتحويل القيمة المخزنة إلى حروف صغيرة أولا ثم تقارن مع الكلمة 'manager' وهي أيضا بالحروف الصغيرة.

```
Oracle SQL*Plus
File Edit Search Options Help
SQL> Select * from Emp
  2  where Lower(job) = 'manager';

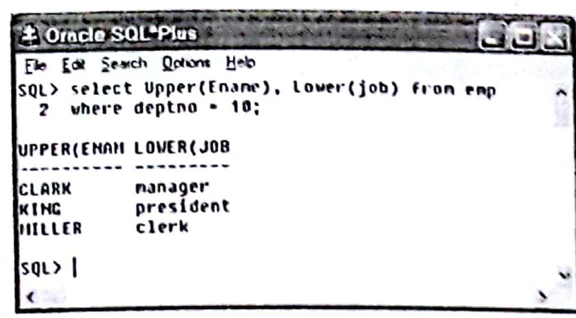
    EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
--------- ---------- --------- --------- --------- --------- --------- ---------
     7566 JONES      MANAGER         7839 02-APR-81      2975                    20
     7698 BLAKE      MANAGER         7839 01-MAY-81      2850                    30
     7782 CLARK      MANAGER         7839 09-JUN-81      2450                    10

SQL>
```

مثال :

لإظهار أسماء الموظفين بالحروف الكبيرة والوظائف بالحروف الصغيرة للعاملين بالقسم رقم 10.

```
select Upper(Ename), Lower(job) from emp
where deptno = 10;
```



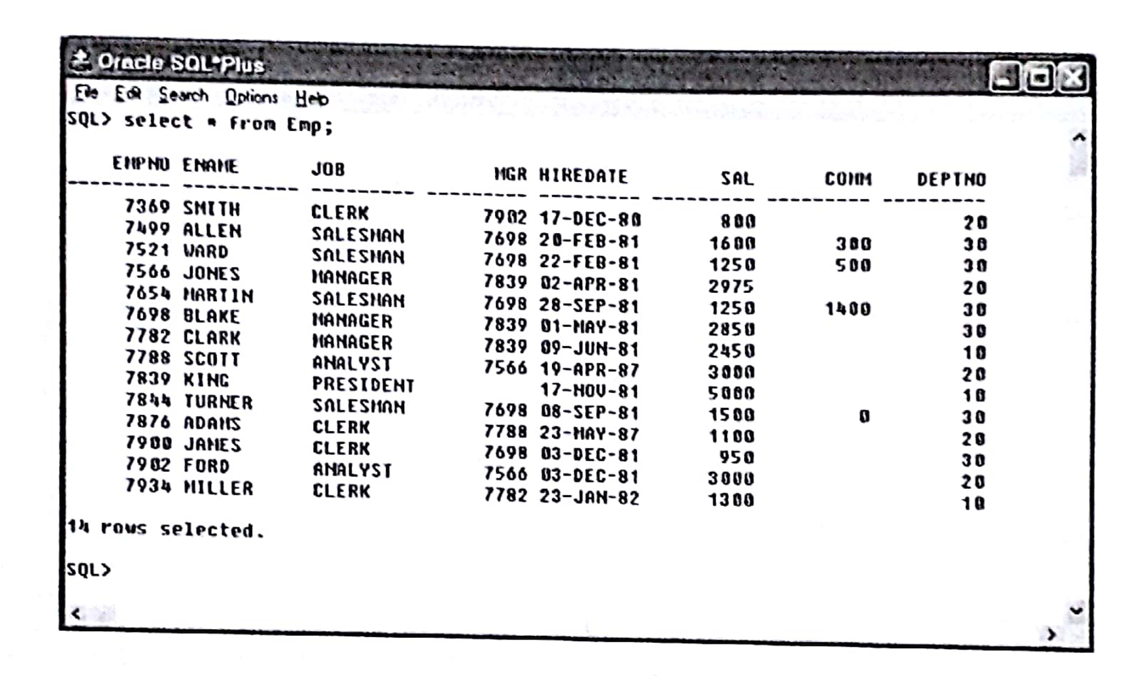
```
Oracle SQL*Plus
File Edit Search Options Help
SQL> select Upper(Ename), Lower(job) from emp
  2  where deptno = 10;

UPPER(ENAM LOWER(JOB
---------- ---------
CLARK      manager
KING       president
MILLER     clerk

SQL>
```

Exercises تمارين

من خلال جدول الموظفين Emp والموضح بالشكل التالي :



Oracle SQL*Plus

File Edit Search Options Help

```
SQL> select * from Emp;
```

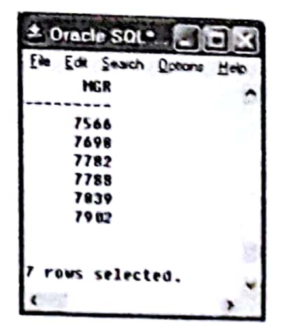
EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

14 rows selected.

SQL>

استرجع السجلات حسب كل تمرين :

✍ (1) استرجع قائمة بكل أرقام المدراء من عمود المدراء ***MGR*** بشكل غير متكرر (مرة واحدة فقط لكل رقم) بحيث تكون النتيجة كالتالي :



Oracle SQL*Plus

File Edit Search Options Help

MGR
7566
7698
7782
7788
7839
7902

7 rows selected.

....

....

....

....

....

....

....

(2) استرجع قائمة بالموظفين العاملين بالقسم رقم 30 مع مرتباتهم و قيمة الضريبة علما بأن الضريبة هي 10% من قيمة المرتب.

Oracle SQL*Plus
File Edit Search Options Help

ENAME	SAL	TAX
ALLEN	1600	160
WARD	1250	125
MARTIN	1250	125
BLAKE	2850	285
TURNER	1500	150
JAMES	950	95

6 rows selected.

SQL>

....
....
....
....
....
....
....

(3) استرجع قائمة بأرقام وأسماء الموظفين ومرتباتهم وأرقام أقسامهم فقط للعاملين بالقسم رقم 10 أو 30 والذين مرتباتهم بين 1000 و 2000.

Oracle SQL*Plus
File Edit Search Options Help

EMPNO	ENAME	SAL	DEPTNO
7499	ALLEN	1600	30
7521	WARD	1250	30
7654	MARTIN	1250	30
7844	TURNER	1500	30
7934	MILLER	1300	10

SQL>

....
....
....
....
....
....
....

(4) استرجع قائمة بأرقام وأسماء الموظفين ووظائفهم لغير العاملين بالقسم رقم 30 وألا يكونوا مدراء.

Oracle SQL*Plus
File Edit Search Options Help

EMPNO	ENAME	JOB
7499	ALLEN	SALESMAN
7521	WARD	SALESMAN
7654	MARTIN	SALESMAN
7844	TURNER	SALESMAN
7900	JAMES	CLERK

SQL>

....
....
....
....
....
....
....

✍ (5) استرجع قائمة بأرقام وأسماء الموظفين والتي تنتهي أسمائهم بـ 'er'

```
Oracle SQL*Plus
File Edit Search Options Help
  3  ;

     EMPNO ENAME
---------- ----------
      7844 TURNER
      7934 MILLER

SQL> |
```

....
....
....
....
....
....
....

✍ (6) اوجد رقم واسم ووظيفة الموظف الذي لم يتم تعيين مدير له (أي لا توجد قيمة له في عمود *Mgr*).

```
Oracle SQL*Plus
File Edit Search Options Help
SQL> /

     EMPNO ENAME      JOB
---------- ---------- ---------
      7839 KING       PRESIDENT

SQL>
```

....
....
....
....
....
....
....

✍ (7) استرجع قائمة بأسماء ووظائف العاملين مرتبة تصاعديا على أساس نوع الوظيفة فقط للعاملين التي تحتوي أسمائهم على حرف 'A'

```
Oracle SQL*P.
File Edit Search Options Help
ENAME      JOB
---------- ---------
ADAMS      CLERK
JAMES      CLERK
BLAKE      MANAGER
CLARK      MANAGER
ALLEN      SALESMAN
WARD       SALESMAN
MARTIN     SALESMAN

7 rows selected.

SQL>
```

....
....
....
....
....
....
....
....
....

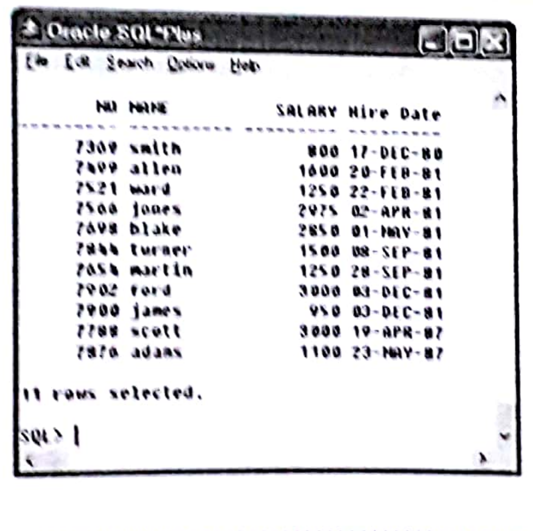
✍ (8) أوجد الفرق بين المتوسط الحسابي للمرتبات واصغر مرتب ، وكذلك المتوسط الحسابي للمرتبات واكبر مرتب للعاملين بالقسم رقم 20.

```
Oracle SQL*Plus
File Edit Search Options Help
SQL> /

  AVG_MIN    AVG_MAX
---------- ----------
      1375       -825

SQL>
```

✍ (9) استرجع قائمة بأرقام وأسماء ومرتبات وتاريخ الاستخدام (*HireDate*) للعاملين بالقسمين 20 و 30 على أن تكون مرتبة تصاعديا على أساس تاريخ الاستخدام وإذا تساوت القيم يتم الترتيب تنازليا على أساس المرتب (الأسماء يجب أن تطبع بالحروف الصغيرة) وعلى أن تظهر أسماء الأعمدة كما بالشكل التالي:



NO	NAME	SALARY	Hire Date
7369	smith	800	17-DEC-80
7499	allen	1600	20-FEB-81
7521	ward	1250	22-FEB-81
7566	jones	2975	02-APR-81
7698	blake	2850	01-MAY-81
7844	turner	1500	08-SEP-81
7654	martin	1250	28-SEP-81
7902	ford	3000	03-DEC-81
7900	james	950	03-DEC-81
7788	scott	3000	19-APR-87
7876	adams	1100	23-MAY-87

11 rows selected.

SQL>

حل التمارين

(1)

```
select distinct mgr from emp;
```

(2)

```
select ename , sal , sal*0.10 as tax from emp
 where deptno = 30;
```

(3)

```
select EmpNo, ename, sal, deptno from emp
where (sal between 1000 and 2000)
and (deptno in (10, 30));
```

(4)

```
select EmpNo, ename, job from emp
where job != 'MANAGER'
and deptno != 30;
```

(5)

```
select empno, ename from emp
where ename like '%ER';
```

(6)

```
select empno, ename, job from emp
where mgr is null;
```

(7)

```
select ename, job from emp
where ename like '%A%'
order by job asc;
```

(8)

```
select avg(sal)- min(sal) as Avg_Min,
       avg(sal) - Max(sal) as Avg_Max from emp
where deptno = 20;
```

(9)

```
select empno as No,
       Lower(ename) as Name ,
       sal as Salary,
       hiredate as 'Hire Date' from emp
where deptno in (20, 30)
order by hiredate asc, sal desc;
```

اعدها : عبد الناصر الغويل